

Fast and Accurate Retinal Vasculature Tracing and Kernel-Isomap-based Feature Selection

Donghyeop Han, Heeyoul Choi, Choonseog Park, and Yoonsuck Choe

Abstract—The blood vessels in the retina have a characteristic radiating pattern, while there exists a significant variation dependent on the individual and/or medical condition. Extracting the geometric properties of these blood vessels have several important applications, such as biometrics (for identification) and medical diagnosis. In this paper, we will focus on biometric applications. For this, we propose a fast and accurate algorithm for tracing the blood vessels, and compare several candidate summary features based on the tracing results. Existing tracing algorithms based on a detailed analysis of the image can be too slow to quickly process a large volume of retinal images in real time (e.g., at a security check point). In order to select good features that can be extracted from the traces, we used kernel Isomap to test the distance between different retinal images as projected onto their respective feature spaces. We tested the following feature set: (1) angle among branches, (2) the number of fiber based on distance, (3) distance between branches, and (4) inner product among branches. Our results indicate that features 3 and 4 are prime candidates for use in fast, realtime biometric tasks. We expect our method to lead to fast and accurate biometric systems based on retinal images.

I. INTRODUCTION

The blood vessels in the retina have a characteristic radiating pattern, while there exists a significant variation dependent on the individual and/or medical condition. Extracting the geometric properties of these blood vessels have several important applications, such as biometrics (for identification) and medical diagnosis. Compared to other types of biometric technology, the blood vessels in the retina have the following benefits. First, the blood vessel pattern in the retina barely changes unless people have some disease such as diabetic retinopathy, glaucoma, cataracts, or other diseases of the eyes [1]. Second, the large, unique features can be extracted from the blood vessel pattern of the retina within a small region, allowing quick identification (for example, 400 data points are used for generating a 96-byte template) [2]. Third, it is the most tamper-resistant biometric due to the fact that the retina is protected in the eye, isolated from the external environment [3]. In this paper, we will focus on biometric applications especially for security and identification systems that require real time processing.

In order to extract the retinal vessels in real time, we propose a fast algorithm which searches branch points as the retinal biometric. Most vascular tracing methods can be categorized into two classes. The first class uses pixel or segmentation-based processing [4]–[8]. They are applied

to every pixel or unit segment on the structures that have complex vessels and abundant branches. Because several operations are required such as template processing and thresholding [9], [10] on every pixel along the edges, it takes a relatively long time to trace the structures. Some methods use a priori knowledge about the structures of interest, which makes the process less general to various conditions [11], [12]. Moreover, most of the methods have difficulty in handling branches, except that Osher et al. introduced level set methods [13] to operate properly on branch points. However, the method needs to compute derivatives at every pixel on the isosurfaces, which makes the algorithm unsuitable for real-time environments.

The second class is based on vector tracing [14]–[17]. These methods use vector-based approaches that are faster compared to the first class of algorithms above due to the fact that every pixel does not need to be checked. However, they face the following general difficulties. The process is not able to continue tracing at branch points or only one branch is selected for continued tracing. This issue makes it hard to extract the branch information. Haris et al. addressed these two issues using a model-based algorithm. However, Haris et al.'s algorithm yields another issue such as backtracking, which can result in incorrect, redundant traces [16].

Motivated by the previous methods, we introduce a new algorithm using Moving Windows (MW) to solve simultaneously the issues of speed, branch handling, and accuracy. The MW approach is robust to noise by use of a local contrast intensity checking of pixels. MW is also able to handle branched vessels within the window based on the parts of the vessel that intersect the edges of the window. Our method uses interpolation to help speed up processing. The interpolation skips a majority of pixels within the window and operates on a small number of sampled pixels-of-interest. It uses control points and the derivatives at the source points, which produces accurate results.

Selecting suitable features from candidate summary features that include many irrelevant or redundant and detailed features could be very important for verifying all classes of retinal vasculature because weak biometric traits show poor discriminating performance during verification [18], [19]. In order to select good features, we checked the distance matrix which can reveal how scattered the features are, and the histogram of the distances which can reveal how the distances are distributed. For this, we used kernel Isomap to test the distance between different retinal images as projected onto their respective feature spaces.

Even though the features are extracted from the images,

Donghyeop Han, Heeyoul Choi, Choonseog Park, and Yoonsuck Choe are with the Department of Computer Science and Engineering, The Texas University of A&M, College Station, Texas, USA. e-mail: dhan1@cs.tamu.edu, hchoi@cs.tamu.edu, cspark13@cs.tamu.edu, and choe@cs.tamu.edu.

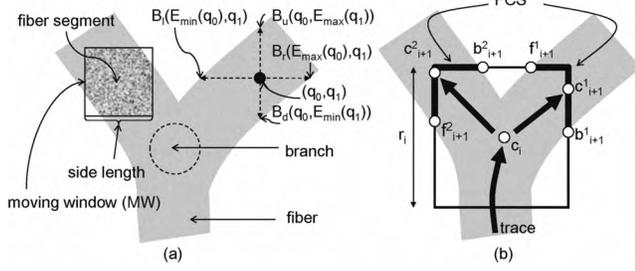


Fig. 1. Notations. (a) Fiber. (b) Moving window.

in order to make the features visible in 2D or 3D space we need to reduce the dimension. Manifold learning involves inducing a smooth nonlinear low-dimensional manifold from a set of data points drawn from the manifold. Though various methods (for example see [20]–[22]) have been developed, in this paper we use kernel Isomap [23]. Kernel Isomap was proposed as a variety of Isomap [21] which is a representative isometric mapping method. Kernel Isomap projects training data points into a low dimensional space in a nonlinear way and can project new data points into the same space. Moreover, the feature space extracted from kernel Isomap is more robust against noise than those from other methods.

The experimental result shows that our tracing algorithm is sufficiently fast enough for real time verification system. Each data image was 150×130 in size. The average and standard deviation of processing time per image was 0.053 seconds and 0.021 seconds respectively. We also obtained 90.78% of tracing accuracy in synthetic images using ground truth. As for the features, we tested the following feature set: (1) angle among branches, (2) the number of fiber based on distance, (3) distance between branches, and (4) inner product among branches. Features 3 and 4 look much better than features 1 and 2 using both the distribution of the distances between all the pairs of points and the two dominant dimensional space of the features. There might be some information loss in features 1 and 2 compared to features 3 and 4.

The remainder of this paper is organized as follows. In Section II, we present our tracing algorithm utilizing MW. In Section III, we describe the selection of kernel-Isomap-based feature. In Section IV, the results of tracing algorithm and kernel-Isomap-based feature extraction are shown and analyzed. In Section V, we discuss issues arising from this work, followed by conclusions in Section VI.

II. MOVING WINDOW TRACING ALGORITHM

Table. I summarizes the terminology and symbols used in this paper. Fig. 2(a) shows the basic process of our algorithm. MW detects fibers and an interpolation curve (Cubic tangential trace splines: CTTS) is used in tracing based on Sun et al.'s algorithm [24].

A. Moving Window (MW)

Our algorithm uses a moving window to keep track of the trace. Initial moving window construction requires a automatic seed point selection from grid lines on image. If

Notation	Meaning	Example
Fiber	The object to be traced.	Fig. 1(a)
Branch	The location where two fibers meet.	Fig. 1(a)
Moving Window (MW)	A rectangular tracing region. MW_i is the i^{th} MW.	Fig. 1(a)
(q_0, q_1)	Seed point coordinate supplied by the automatic detection.	Fig. 1(a)
B_r, B_l, B_u, B_d	The right, left, upward, and downward object/background boundary points relative to (q_0, q_1) , respectively.	Fig. 1(a)
FCS	Fiber's cross section on the edges of the MW. FCS_{i+1}^n is the n^{th} FCS of MW_i .	Fig. 1(b)
source, target	c_i is the source point of MW_i . c_{i+1}^n is the n^{th} target point of MW_i .	Fig. 1(b)
flanking points	f_{i+1}^r is the left end point of FCS_{i+1}^n . b_{i+1}^l is the right end point of FCS_{i+1}^n .	Fig. 1(b)
$\ x, y\ $	The distance between x and y along the FCS.	Eq. 6

TABLE I

NOTATION USED IN THE PAPER.

the lowest intensity pixel on each grid line is lower than α and the variance of intensity values on the line is higher than β , the pixel is considered as a seed point (α and β were set to 112 and 5 respectively in our experiment). In order for our algorithm to trace medial axis, the arbitrary seed point should be relocated to the closest point on the medial axis by the following equations:

$$q'_j = q_j + QE(q_j), \quad j = 0, 1 \quad (1)$$

where (q_0, q_1) is the auto-detected seed point, (q'_0, q'_1) is the adjusted seed point, and QE is the quantization error between (q_0, q_1) and (q'_0, q'_1) from Eq. 2-4. The distance d_k is defined as follows:

$$d_k = \begin{cases} E_{max}(q_{k/2}) - q_{k/2}, & \text{for } k \% 2 = 0 \\ E_{min}(q_{k/2}) - q_{k/2}, & \text{for } k \% 2 = 1, \end{cases} \quad (2)$$

where $k \in \{0, 1, 2, 3\}$ and $\%$ the modulus. Four boundary points ($B_r, B_l, B_u,$ and B_d shown in Fig. 1(a)) are obtained by a moving Gaussian filter from (q_0, q_1) to the right, left, up, and down, respectively. We simply use 2D Gaussian filter for detecting the four boundary points based on the difference between the intensity of (q_0, q_1) and the filter's response. $E_{max}(q_0)$ is the x coordinate of B_r , $E_{min}(q_0)$ is the x coordinate of B_l . $E_{max}(q_1)$ and $E_{min}(q_1)$ are the y coordinates of B_u and B_d respectively as shown in Fig. 1(a).

$$D = \operatorname{argmin}_{k \in \{0,1\}} (|d_{2k}| + |d_{2k+1}|) \quad (3)$$

$$QE(q_j) = \begin{cases} (d_{2D} + d_{2D+1})/2, & D = j \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

From the quantization error (QE), adjusted seed point (q'_0, q'_1) can be obtained as shown in Fig. 2(b). The creation of moving window centered at (q'_0, q'_1) also needs an appropriate determination of the window's side length r :

$$r = 2(|d_{2D}| + |d_{2D+1}|) \times \epsilon \quad (5)$$

where ϵ is a scaling factor usually set between 1.1 and 1.5 based on our experimental results.

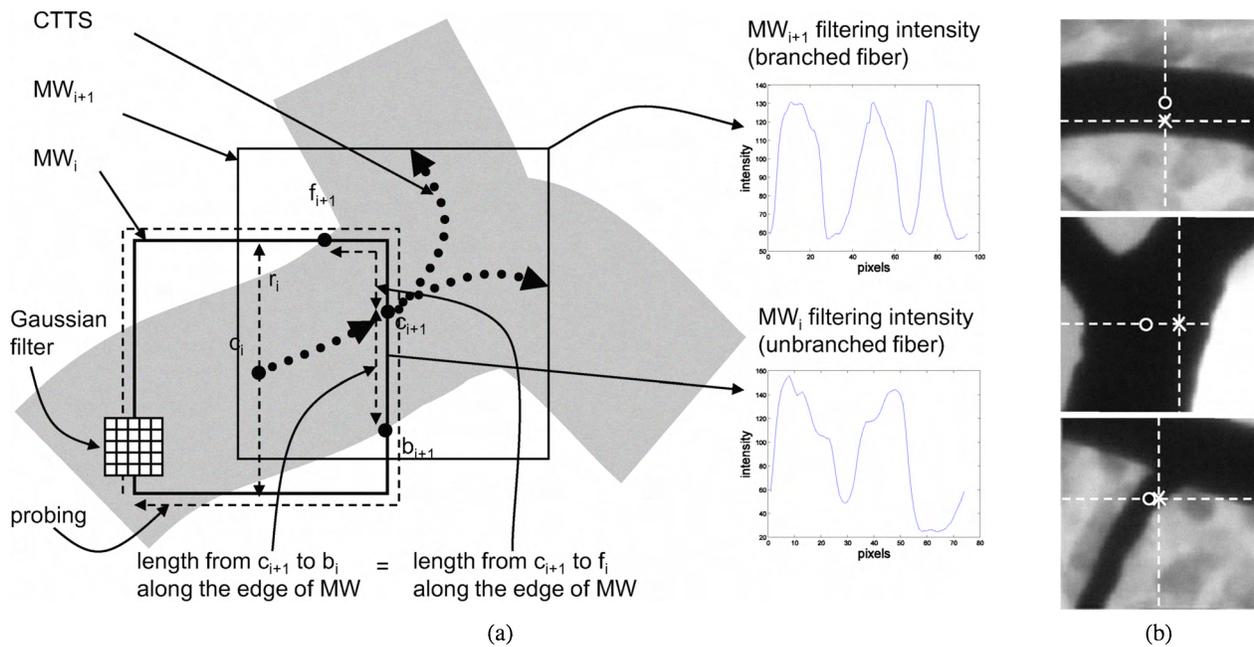


Fig. 2. (a) Overview of our algorithm. (b) Seed point adjustment. Auto-detected and adjusted seed points are marked with \times and \circ , respectively. Dashed lines are grid lines.

After constructing the MW, the edge of the MW is probed to detect the fiber's cross section with the MW (FCS) as shown in Fig. 1(b). A Gaussian filter is applied along the MW border (Fig. 2(a), left), resulting in a pixel intensity profile sampled only near the MW border (Fig. 2(a), right). The probing on the MW is detailed in Section II-B. As we can see, branched fibers have three or more valleys whereas unbranched ones have only two valleys (fibers are assumed to be dark). The centers of the detected FCS are located on the border of the MW. Fig. 2(a) shows that the target c_{i+1} must have the same distance from the flanking points f_{i+1} , b_{i+1} (defined in Table. I) of the FCS_{i+1} and must belong to the FCS_{i+1} . In subsequent steps, newly generated MWs are given a different side length, based on the current fiber width:

$$r_{i+1} = 2\|f_{i+1}, c_{i+1}\| \times \epsilon \quad (6)$$

where ϵ is the same as Eq. 5, r_{i+1} is the next MW side length. The FCS which contains the previous trace result does not generate a new MW in order to avoid backtracking. The centers of the detected FCS are often not exactly located on the medial axis. There are many effective methods to get a point on the medial axis [25]–[27]. However, these methods require expensive computation using the gradient value, Hessian matrix, or level-set-based-framework. In this paper, we use the simple method above to get the centers of the detected FCS, which has linear processing time. Moreover, it can handle the backtracking problem, which cannot be solved by existing methods [16], [17].

B. Fiber Detection with Local Contrast

In many cases, biomedical images have locally low or high contrast due to uneven illumination, noise, and lighting

variation [28]–[30]. Even if trace algorithms start from a seed point in a high contrast area, the trace can move in to a low contrast area. In this case, we cannot use global threshold. Therefore, in order to find fibers in both low and high contrast areas, we use FCS measurement, $F_m(x)$, defined as

$$F_m(x) = \exp\left(-\frac{(I(x) - L_{max})^2}{2(0.5 \times k)^2}\right) \quad (7)$$

where k and L_{max} are the local contrast and the local maximum intensity value of the edge of the current MW. $I(x)$ is the intensity value at x located on the edge of the MW. If $F_m(x)$ is greater than a predefined threshold, x is considered to be a member of FCS. The empirically predefined threshold is usually set to 0.4. The k , L_{max} , and $I(x)$ are sensitive to high intensity noise when a dark image has low contrast. In order to avoid this problem, we use a 5×5 Gaussian filter to get k , L_{max} , and $I(x)$. $F_m(x)$ in Eq. 7 is for dark background image. However, $F_m(x)$ can be used also for bright background image after switching L_{max} to L_{min} which is the local minimum intensity value of the edge of the current MW. Using $F_m(x)$, we can trace noisy and low contrast area from one seed point on high contrast area without setting any parameters.

C. Stopping Criteria

The tracing process is terminated if one or more of the following conditions are satisfied.

- 1) The newly generated MW is outside the image data.
- 2) The number of FCSs detected on the current MW is 1, which means that the fiber is traced to its end.
- 3) Local contrast in Eq. 7 is less than a predefined threshold (usually set to 10 which is less than 30

percent of the gray level contrast that human eye can distinguish). Small threshold may cause the detection of many false positives, while large threshold may cause early termination of tracing. In our experiments, the threshold value of 10 gave the best results even in the image where the human eye cannot easily differentiate fibers.

- 4) A previously detected fiber intersects the current MW. All the pixels on FCSs detected on the current MW are checked for this test detailed in Section II-D.

D. Connection between Traced Results

CTTS curve is a nonlinear interpolation with control points using the modified Sun et al.'s algorithm [24]. We extract control points proportional to the distance between center points of each tracing step while Sun et al.'s algorithm generates control points proportional to fiber width. If a fiber's intensity is corrupted by image noise, either the termination of tracing can occur or branch may not be detected. In this case, the traced fibers' intensities are a key factor to measure the likelihood of connection. The following algorithm can cover all the cases with the help of MW-CTTS. Let $I_{\{1..4\}}$ denote the intensity values of the cross points between the current MW and the previously traced result. These intensity values are used in order to determine the connection between the cross points and the current MW's center whose intensity value is I_c . In order to use these intensity values, we define the following three terms. Let $IC(t, x, y)$ denotes an intensity of a t point located on a CTTS curve whose endpoints are x and y . In addition to this, $IM(x, y) = ((I_x + I_y)/2)$ and $IK(x, y) = ((I_x - I_y)/2)$ are described to represent the average and the half-difference of the two intensity values. For each cross point, we interpolate a CTTS curve and check connection measurement, $C_m(j)$, defined as

$$C_m(j) = \int_c^j \exp\left(-\frac{(IC(t, c, j) - IM(j, c))^2}{2IK(c, j)^2}\right) dt/n \quad (8)$$

where j is a cross point, c is the center point of the current MW, and n is the length of the CTTS curve from c to j . Hence, $\int_c^j IC(t, c, j) dt$ denotes summation of intensity values along the CTTS curve from c to j . Among the cross points, we select one point with the $C_m(l)$ obtained from

$$l = \operatorname{argmax}_{j \in [1, \dots, s]} C_m(j) \quad (9)$$

where s is the number of the cross points. If $C_m(l)$ is larger than a predefined threshold which is set to 0.2, CTTS from c to l is drawn.

E. Features

From the tracing result of retinal blood vessels, we can extract the branch points as retinal feature. Each branch point has its own number of members in the FCS set described in Eq. 7. The higher the number of members in the FCS set, the more likely that the branch appears clearly. We further sort branches based on the number in decreasing order. From the

high ranked branches, we define the following four features assuming that B_i denotes the i^{th} ranked branch point (Fig. 3).

- 1) Angle among branches: The first three ranked branches (B_1, B_2 , and B_3) can make a triangle. From the highest ranked branch to the lowest ranked branch, we can measure three angles such as $\angle B_3 B_1 B_2$, $\angle B_1 B_2 B_3$, and $\angle B_2 B_3 B_1$. The next triangle can be generated by B_2, B_3 , and B_4 . This triangle yields three angles in the same way as the previous triangle. This feature is based on the curvature of features [31].
- 2) The number of fiber based on distance: At each branch, we made three circles whose radius were 5, 7, and 9 respectively. We probed each circle's edge to find the number of fibers crossed on the edge using Eq. 7. Each branch generates three elements of this feature based on the radius [32].
- 3) Distance between branches: Like the first feature, we use triangles for square root of Euclidean distance between branches like $\sqrt{B_1 B_2}$, $\sqrt{B_2 B_3}$, and $\sqrt{B_3 B_1}$. The way to make triangles is the same as the first feature. The definition of this feature comes from the length of features [31].
- 4) Inner product among branches: From the three highest ranked branches, we can get the first three inner products ($B_1 B_3 \bullet B_1 B_2$, $B_2 B_1 \bullet B_2 B_3$, and $B_3 B_2 \bullet B_3 B_1$). We used square root of four of dot product for normalization. In similar way, the next ranked branches yield inner products [33].

III. ANALYSIS OF FEATURES USING KERNEL ISOMAP

Given the feature set, we need to compare the quality of the features. To do so, we check the distances of features and make the features visible in 2D or 3D space. To see how the features are distributed, the histogram of the distances of data points from each feature are useful. Here, we used the Mahalanobis distance because each element of the features has different unit compared to others.

We use kernel Isomap to see how good the features are in a low dimensional space. In this section we review kernel Isomap briefly [23]. Given N objects with each object being represented by an m -dimensional vector \mathbf{x}_i , $i = 1, \dots, N$, the kernel Isomap algorithm finds an implicit mapping which places N points in a low-dimensional space. In contrast to Isomap, the kernel Isomap can project novel data points onto the discovered low-dimensional space, as well, through a kernel trick. The kernel Isomap mainly exploits the solution of the additive constant problem, the goal of which is to find an appropriate constant to be added to all dissimilarities (or distances), apart from the self-dissimilarities, that makes the kernel matrix to be positive semidefinite.

Given a distance matrix, we calculate Dijkstra's geodesic distances (shortest paths) \mathbf{D} , and calculate the doubly centered kernel matrix as below.

$$\mathbf{K} = -\frac{1}{2} \mathbf{H} \mathbf{D}^2 \mathbf{H}, \quad (10)$$

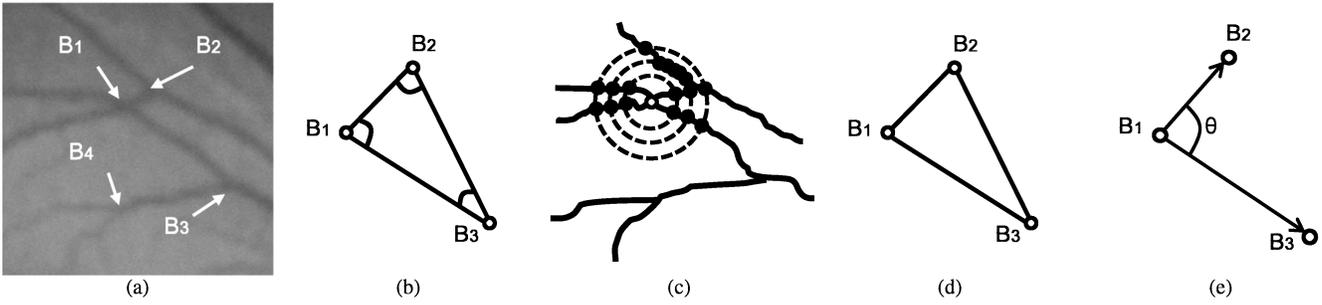


Fig. 3. (a) Ranked branches in an original image. (b) Feature 1: angles between branches. (c) Feature 2: the number of fibers (black discs) around a central branch point (white disc) computed based on distance (concentric circles shown with dashed lines). (d) Feature 3: length of lines between branches. (e) Feature 4: inner products among branches.

where $\mathbf{D}^2 = [D_{ij}^2]$ means the element-wise square of the geodesic distance matrix $\mathbf{D} = [D_{ij}]$, \mathbf{H} is the centering matrix, given by $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{e}_N\mathbf{e}_N^T$ for $\mathbf{e}_N = [1 \dots 1]^T \in \mathbb{R}^N$. Then, we make the kernel matrix positive definite by adding a constant, c .

$$\tilde{\mathbf{K}} = \mathbf{K}(\mathbf{D}^2) + 2c\mathbf{K}(\mathbf{D}) + \frac{1}{2}c^2\mathbf{H}, \quad (11)$$

where c is the largest eigenvalue of the matrix

$$\begin{bmatrix} \mathbf{0} & 2\mathbf{K}(\mathbf{D}^2) \\ -\mathbf{I} & -4\mathbf{K}(\mathbf{D}) \end{bmatrix}. \quad (12)$$

Eq. 11 implies substituting $\tilde{\mathbf{D}}$ for \mathbf{D} in Eq. 10, which is given by

$$\tilde{D}_{ij} = D_{ij} + c(1 - \delta_{ij}), \quad (13)$$

which makes the matrix \mathbf{K} to be positive semi-definite. The term δ_{ij} is the Kronecker delta. Finally, projection mapping \mathbf{Y} is given by Eq. 14 after eigen-decomposition, $\tilde{\mathbf{K}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$.

$$\mathbf{Y} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}}. \quad (14)$$

The projection for novel data points is similar to kernel PCA and is described in [23].

IV. EXPERIMENTS AND RESULTS

A. Tracing Result

Fig. 4 shows that the tracing result of retinal vasculature. The original images were downloaded from the STARE (Structured Analysis of the Retina) website [34]. Each image has the size of 150×130 . We traced 89 images from the database. We performed the experiment on a PC with Intel Pentium 4 (2.4 GHz) processor, 512MB of memory under Windows XP operating system. C/C++ programming language and OpenGL graphics library were used. The average and standard deviation of processing time per image was 0.053 seconds and 0.021 seconds respectively, which means that this tracing algorithm is fast enough for real time identification systems. However, there were false positives around the optic disc (white arrows in Fig. 4). The optic disc appears as a white region in retinal images [35]. This white region can make our tracing algorithm incorrectly mark part of the background as vasculatures around the optic disc. It is due to the fact that the fiber detection uses the information

of local contrast (Section II-B). We will handle this problem in our future work, by identifying the optic disc [36], [37].

We analyze how efficiently the proposed method is able to process large amounts of data. Assuming that one fiber, whose length is k , is located on a MW whose side length is $2n\epsilon$, where n is the width of the fiber, we can evaluate the required number of pixels to be processed by the sum of the size of Gaussian filter, the MW side length, and the number of MW. The number of pixels to be processed by MW, denoted P_{MW} , is then calculated as follows: $P_{MW} = (2n\epsilon \times 4) \times (5 \times 5) \times \frac{k}{2n\epsilon} = 100 \times k$ where $2n\epsilon \times 4$ is the MW side length, 5×5 is the filter size, and $\frac{k}{2n\epsilon}$ is the total number of MWs. Consequently, the proposed trace has $O(k)$ (i.e., linear) processing time, depending only on the fiber length (i.e., not all pixels in the fiber), which indicates that the proposed method is also efficient for large-scale data sets without additional overhead.

We conducted an experiment where we traced the vasculature data with our and other well known vector based tracing algorithms for speed performance comparison (table. II). Our method (MW-CTTS) turned out to be the fastest and the processing time is short enough for real time systems. Can et al.'s method uses many templates in each trace step, which makes the trace slower than the others. Haris et al.'s method does not trace to the edge of its circular window template in each trace step, which makes this trace slower than MW-CTTS. Note that MW-CTTS has the largest total traced distance starting from a single seed point, which gives more complete trace than the others. From this experiment, we also calculated the total processing time for complete traces of the vasculature data (#6 in table. II) using more than one seed point. The number of total traced pixels was 9501. For the complete traces, Can et al.'s method and Haris et al.'s method took $2177.70 \mu\text{s}$ and $1682.94 \mu\text{s}$ respectively. Our proposed method took $1233.54 \mu\text{s}$.

We also conducted another experiment using synthetic data for accuracy. We generated three 256×256 pixel datasets. Each dataset contained linear, curvy, and spiral fiber, respectively. Generally, lines and curves most frequently appear in blood vessel data. The spiral data contains curvatures ranging from 0.17 to 0.0044 to cover most vessels. The numbers of ideal centerline points are 200, 230, and 2900 for line, curve, and spiral data. The fiber's radius ranges from 2.0 to 4.0 pixels and the background intensity is 0. All images have

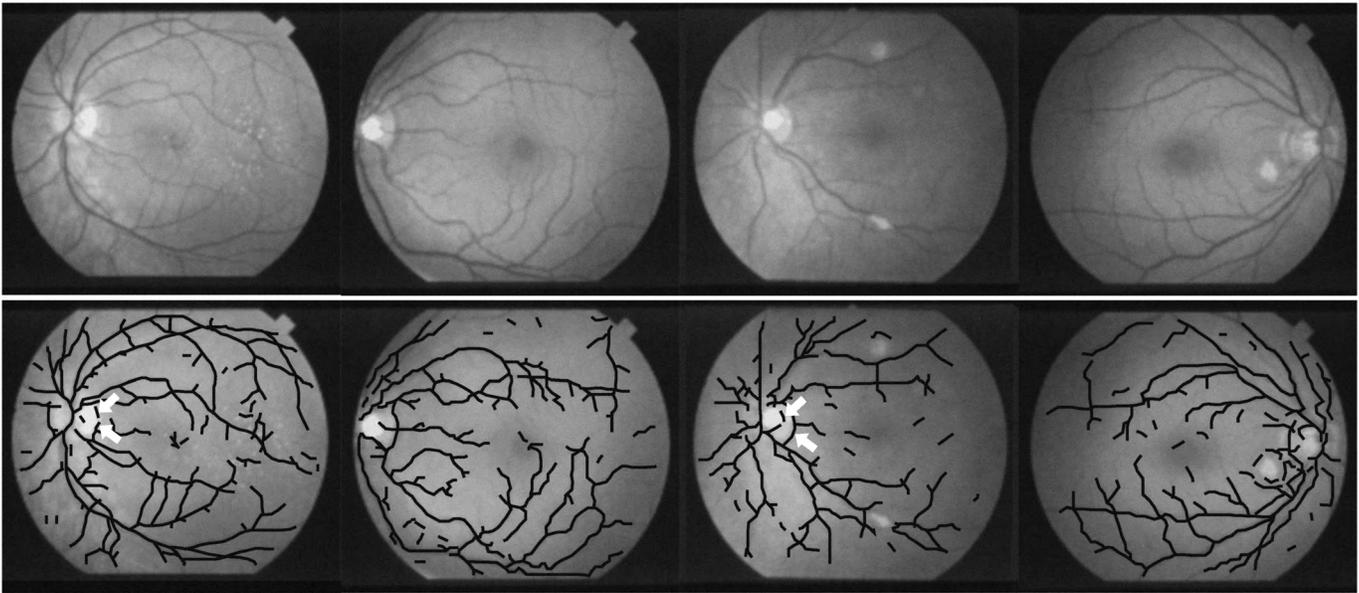


Fig. 4. Example tracing results from retina vasculatures. Each image was 150×130 in size. The average and standard deviation of processing time per image was 0.053 seconds and 0.021 seconds respectively. Top row: original images. Bottom row: tracing results (black traces). False detections around optic disc (white arrows).

	Can et al.'s method		Haris et al.'s method		MW-CTTS	
	total traced distance	$\mu\text{sec}/\text{distance}$	total traced distance	$\mu\text{sec}/\text{distance}$	total traced distance	$\mu\text{sec}/\text{distance}$
#1	68 pixels	0.50211	831 pixels	0.08791	936 pixels	0.06706
#2	25 pixels	1.00660	1826 pixels	0.16842	1912 pixels	0.12579
#3	76 pixels	0.85746	1473 pixels	0.22864	7845 pixels	0.09975
#4	272 pixels	0.14079	2190 pixels	0.12160	3712 pixels	0.11607
#5	196 pixels	0.16770	1461 pixels	0.14551	4045 pixels	0.10914
#6	216 pixels	0.20837	1449 pixels	0.16103	7731 pixels	0.11803

TABLE II

PERFORMANCE COMPARISON USING ONE SEED POINT ON 6 VASCULATURE DATA. THE $\mu\text{M}/\text{PIXEL}$ IS 1.1.

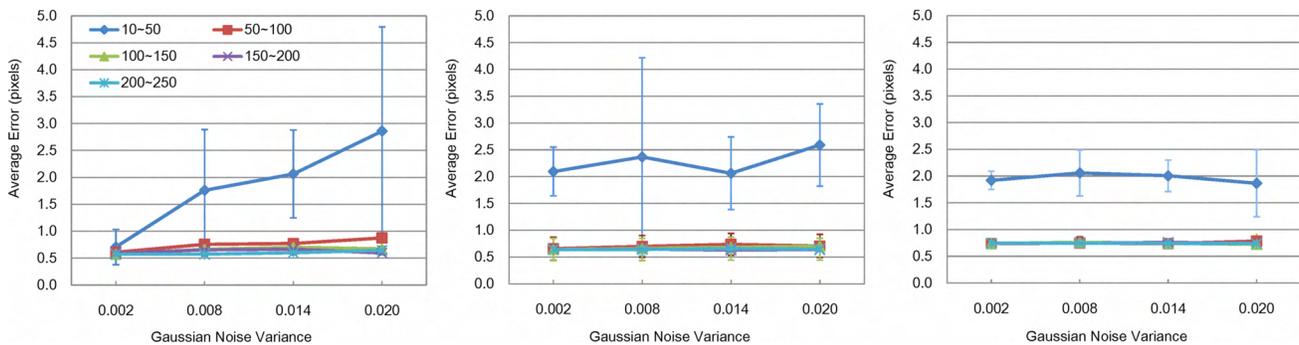


Fig. 5. Means and standard deviations of the Monte Carlo measures for average errors (pixels). Each series represents fibers' different intensity profiles (10~50, 50~100, 100~150, 150~200, and 200~250). Left: Line data. Middle: Curve data. Right: Spiral data.

one of the combinations of fiber's intensity profiles (10~50, 50~100, 100~150, 150~200, and 200~250) and Gaussian noise (0.002, 0.008, 0.004, and 0.020). This synthetic data experiment was conducted to show noise robustness of the proposed algorithm. The performance in Fig. 5 shows the following conclusion. All experiments are robust to the noise representative of medical images, except the lowest intensity data whose intensity profile ranges from 10 to 50. The lowest intensity data has low contrast below the contrast range

humans can differentiate. Except the lowest one, all the other data showed that average error was less than 1 pixel.

B. Feature Comparison

Given the four features, we compared the retinal data sets in two ways: (1) the distribution of the distances between all the pairs of points, and (2) the two dominant dimensional space of the features. The distance distribution can show how the features are scattered in the feature space. The

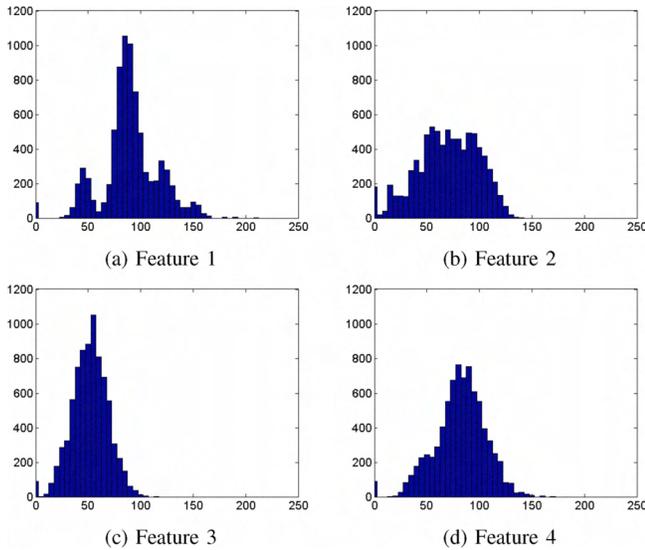


Fig. 6. Histograms of approximated geodesic distances of feature set: (a) Feature 1, (b) feature 2, (c) feature 3, and (d) feature 4. (c) looks similar to (d) while (a) looks similar to (b). (a) and (b) have a few points which are far from other points while (c) and (d) have well distributed distances which is desirable.

two dominant dimensional space for each feature driven by kernel Isomap can show how the feature set actually lies in a low-dimensional Euclidean space converted from features on Riemannian space.

Fig. 6 shows the distribution of the approximated geodesic distances between all the pairs of features. We can tell that in feature 1, there are few points that are far from other points (looking like outliers), and in feature 2, there are some completely overlapped samples. Also, in feature 2, some samples are too close to other samples, while in other features all samples have neighbors a little far from itself. Note that features 3 and 4 are well distributed according to the distances.

In Fig. 7, we can see that features 1, 3 and 4 are more desirable than feature 2. In this figure, feature 1 looks as fine as features 3 and 4. However, in Fig. 8, we can see a difference. The figure shows the 17th and 18th dominant dimensions of the embedded manifold of feature 1. Here, the 18th dimension does not look like a good feature. Actually, when we run a linear method, that is multidimensional scaling (MDS), the first dominant dimension is exactly like this 18th dimension here. Even though kernel Isomap presents better feature space than MDS does, still the distortion remains in some of the feature dimension. Other features have similar 17th and 18th figures as their two leading components have.

From Figs. 6, 7 and 8, features 3 and 4 look better than features 1 and 2. However, it is hard to tell which one among features 3 and 4 is better. The features 3 and 4 are constructed based on distance and inner product, respectively. Usually any distance can be converted into inner product and vice versa. So, basically they have the same amount of information which is interchangeable. This might be the reason why the two features 3 and 4 are very alike. On the other hand, the feature 1 is made of angles which can be

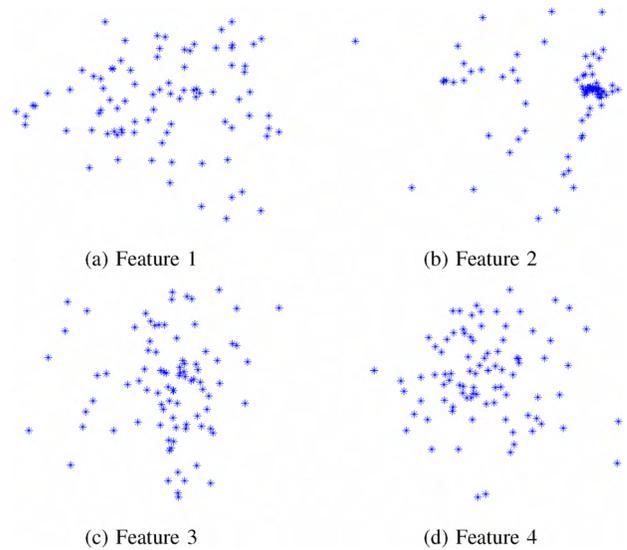


Fig. 7. The first two leading components of the feature set: (a) Feature 1, (b) feature 2, (c) feature 3, and (d) feature 4. As in Fig. 6, (c) looks similar to (d) while (a) looks similar to (b). (a) and (b) have two points which are far from other points while (c) and (d) have well distributed features which is more desirable.

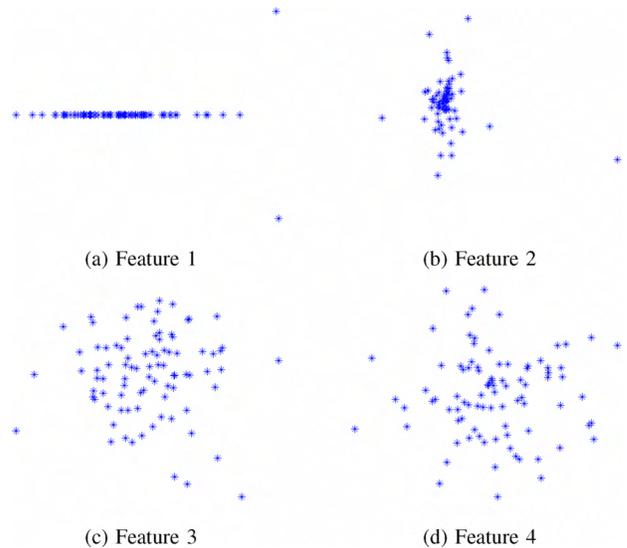


Fig. 8. The 17th and 18th leading components of the feature set: (a) Feature 1, (b) feature 2, (c) feature 3, and (d) feature 4.

obtained by inner product of normalized branches, which means information (the length) is lost, thus it can explain the concentration of the points on 1D in Fig. 8. Feature 2 uses the radius which is related to the length of branches. The radius is discrete so that there might be some information loss.

V. DISCUSSION

The main contribution of the proposed method is that it allows the rapid and accurate extraction of information from retinal blood vessels, making it suitable for security systems. Each retinal image can be easily identified by its unique pattern of blood vessels using features such as branch information. In addition to the security applications, it can support a medical imaging system with a variety of medical

images such as retinal vessel images; blood vessels in MRA, CT, and ultrasound; and nerves and arteries in microscopy.

Another contribution is that kernel Isomap makes it possible to easily see which features are more effective than others using the distances between all the pairs of projected points. As mentioned earlier, kernel Isomap can be considered as a feature extractor for the features 3 or 4. Then, given a new image, we can extract features 3 or 4 and kernel Isomap can project it into the same low dimensional space as the training images.

To make the tracing algorithm fast, we did not use either template matching or knowledge-based approach. Therefore, if some patients have an eye disease such as diabetic retinopathy, glaucoma, or cataract, his/her feature may not be extracted correctly. To overcome this problem, as a future work, we need to consider having simple and general patterns of disease (such as big red oval) so that the proposed method quickly find the disease area before extracting features.

VI. CONCLUSION

In this paper, we have presented a rapid and noise-robust trace method based on MW-CTTS. Experimental results on retinal vasculatures have shown that it is fast enough for security systems and can accurately extract features. Among the extracted features, we showed using kernel Isomap which features are prime candidates for use in fast, realtime biometric tasks. We expect our framework to be more widely used in biomedical systems and engineering applications.

ACKNOWLEDGEMENT

This research was funded in part by NIH/NINDS grant #1R01-NS54252 (PI: Yoonsuck Choe) and Samsung. We would also like to thank the anonymous reviewers for constructive comments.

REFERENCES

- [1] J. GJ. and C. J., "Optical coherence tomography to detect and manage retinal disease and glaucoma," *Am J Ophthalmol*, vol. 69, pp. 137–156, 2004.
- [2] S. T., "Biometric security systems: The best infosec solution?" *ED-PACS*, vol. 30, pp. 1–24, 2003.
- [3] D. K. and M. Grgic, "A survey of biometric recognition methods," *46th International Symposium Electronics in Marine*, pp. 16–18, 2004.
- [4] M. Descoteaux, L. Collins, and K. Siddiqi, "Geometric flows for segmenting vasculature in MRI: Theory and validation," *MICCAI*, pp. 500–507, 2004.
- [5] O. Friman, M. Hindennach, and H. Peitgen, "Template-based multiple hypotheses tracking of small vessels," *IEEE ISBI*, pp. 1047–1050, 2008.
- [6] A. Frangi, W. Niessen, K. L. Vincken, and M. A. Viergever, "Multi-scale vessel enhancement filtering," *MICCAI*, pp. 130–137, 1998.
- [7] W. Niessen, "Model-based image segmentation for image-guided interventions," in *Image-Guided Interventions*. Springer US, 2008.
- [8] H. Li and A. Yezzi, "Vessels as 4d curves: Global minimal 4d paths to extract 3d tubular surfaces," *Proc. IEEE Conference on CVPR*, 2006.
- [9] R. Polli and G. Valli, "An algorithm for real-time vessel enhancement and detection," *Comput. Math. Programs Biomed.*, pp. 1–22, 1997.
- [10] W. K. Pratt, "Digital image processing," *New York: Wiley*, 1978.
- [11] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1316–1323, 2000.
- [12] S. Jehan-Besson, A. Herbulot, M. Barlaud, and G. Aubert, "Shape gradient for image and video segmentation," in *Mathematical Models in Computer Vision: The Handbook*. Springer, 2005.
- [13] S. Osher and N. Paragios, "Geometric level set methods in imaging, vision and graphics," *Springer-Verlag*, 2003.
- [14] K. Al-Kofahi, S. Lasek, S.D., C. Pace, G. Nagy, J. N. Turner, and B. Roysam, "Rapid automated three-dimensional tracing of neurons from confocal image stacks," *IEEE Trans. Inform. Technol. Biomed.*, pp. 171–187, June 2002.
- [15] A. Can, H. Shen, J. Turner, H. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Trans. Inform. Technol. Biomed.*, vol. 3, pp. 125–138, June 1999.
- [16] K. Haris, S. Efstratiadis, N. Maglaveras, C. Pappas, J. Gourassas, and G. Louridas, "Model-based morphological segmentation and labeling of coronary angiograms," *IEEE Trans. Med. Imag.*, vol. 18, pp. 1003–1015, October 1999.
- [17] J. Carrillo, M. Hoyos, E. Davila, and M. Orkisz, "Recursive tracking of vascular tree axes in 3D medical images," *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, pp. 331–339, 2007.
- [18] L. W. James, "Fundamentals of biometric authentication technologies," *International Journal of Image and Graphics (IJIG)*, vol. 1, no. 1, pp. 93–113, January 2001.
- [19] R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos, "Biometric identification through hand geometry measurements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1168–1171, October 2000.
- [20] H. S. Seung and D. D. Lee, "The manifold ways of perception," *Science*, vol. 290, pp. 2268–2269, 2000.
- [21] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
- [22] L. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, June 2003.
- [23] H. Choi and S. Choi, "Robust kernel Isomap," *Pattern Recognition*, vol. 40, no. 3, pp. 853–862, March 2007.
- [24] Y. Sun, "Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm," *IEEE Trans. Med. Imag.*, vol. 8, pp. 78–88, March 1989.
- [25] S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux driven automatic centerline extraction," *Medical Image Analysis*, 2004.
- [26] Hassouna, M.S., Farag, and A.A., "Robust centerline extraction framework using level sets," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 458–465, 2005.
- [27] M. Sofka and C. V. Stewart, "Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures," *IEEE Trans. Med. Imaging*, vol. 25, pp. 1531–1546, 2006.
- [28] L. Dzung, X. Chenyang, and L. Jerry, "A survey of current methods in medical image segmentation," Department of ECE, Johns Hopkins Univ., Tech. Rep., 1999.
- [29] I. Bollmann, P. Quinn, and M. Vela, "Automated particle analysis: Calcareous microfossils," in *Image Analysis, Sediments and Poleoenvironments*. Kluwer, Netherlands: Springer, 2003.
- [30] F. W.-M. Leong, M. Brady, and J. McGee, "Correction of uneven illumination (vignetting) in digital microscopy images," *J. Clin. Pathol*, vol. 56, pp. 619–621, 2003.
- [31] Y. W. Yun, "The 123 of biometric technology," *Synthesis Journal*, pp. 83–96, 2002.
- [32] H. R.B., "Apparatus and method for identifying individuals through their retinal vasculature patterns," *US Patent No. 4109237*, 1978.
- [33] Y. Bulatov, S. Jambawalikar, P. Kumar, and S. Sethia, "Hand recognition using geometric classifiers," *ICBA Int. Conf. Bioinformatics and its Applications*, pp. 16–19, 2004.
- [34] Stare project website. [Online]. Available: <http://www.ces.clemson.edu/ahoover/stare>
- [35] T. Walter, J. Klein, P. Massin, and A. Erginay, "A contribution of image processing to the diagnosis of diabetic retinopathy detection of exudates in color fundus images of the human retina," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1236–1243, 2002.
- [36] F. Mendels, C. Heneghan, and J. Thiran, "Identification of the optic disk boundary in retinal images using active contours," *Irish Machine Vision Image Processing Conference*, pp. 103–115, September 1999.
- [37] K. Akita and H. Kuga, "A computer method of understanding ocular fundus images," *Pattern Recog.*, vol. 15, no. 6, pp. 431–443, 1982.