
Automated Model-Based Segmentation, Tracing, and Analysis of Retinal Vasculature from Digital Fundus Images

Kenneth H. Fritzsche, Ali Can, Hong Shen, Chia-Ling Tsai, James N. Turner, Howard L. Tanenbaum, Charles V. Stewart, and Badrinath Roysam

CONTENTS

6.1	Overview	225
6.2	Introduction	226
6.3	Retinal Imaging Background	233
6.4	Models for Detecting Retinal Vasculature in Digital Imagery	235
6.5	Vessel Extraction Approaches	246
6.6	RPI-Trace	256
6.7	Algorithm Design Considerations for Real-Time Tracing	261
6.8	Accurate Extraction of Vessel Bifurcations and Crossovers	267
6.9	Applications of Vessel Segmentation Data	275
6.10	Implementation Methods	275
6.11	Experimental Validation Using Ground Truth Data	277
6.12	Experimental Analysis of Model and Settings for RPI-Trace	284
6.13	Experimental Assessment of the Impact of Landmark Refinement with the ERPR Algorithm	286
6.14	Chapter Summary	288
	Acknowledgments	289
	References	290

6.1 Overview

Quantitative morphometry of the retinal vasculature is of widespread interest, directly for ophthalmology and indirectly for diseases involving structural and/or functional changes of the body vasculature. Key points such as bifurcations and crossovers are of special interest to developmental biologists

and clinicians examining conditions such as hypertension and diabetes. Segmentation/tracings of the retinal vasculature and the key points, such as bifurcations and crossovers, are also important as spatial landmarks for image registration. Image registration, in turn, has direct applications to change detection, mosaic synthesis, real-time tracking, and real-time spatial referencing. Change detection is important for supporting a variety of clinical trials, high-volume reading centers, and for large-scale screening applications.

The best-available algorithms for segmenting/tracing retinal vasculature are model based, and a variety of models are in use. Depending on the intended application, different algorithmic and implementation choices can be made. This chapter describes some of these models and algorithms and illustrates some of the implementation choices that need to be considered using a real-time algorithm as an example. Also described are methods for extracting key points and a discussion of how vessel morphometric data can be applied. Some methods are presented for generating ground truth or “gold standard” images as well as comparing these against computer-generated results. Finally, some experimental analysis is presented for RPI-Trace and the ERPR landmark determination algorithms developed by our group.

6.2 Introduction

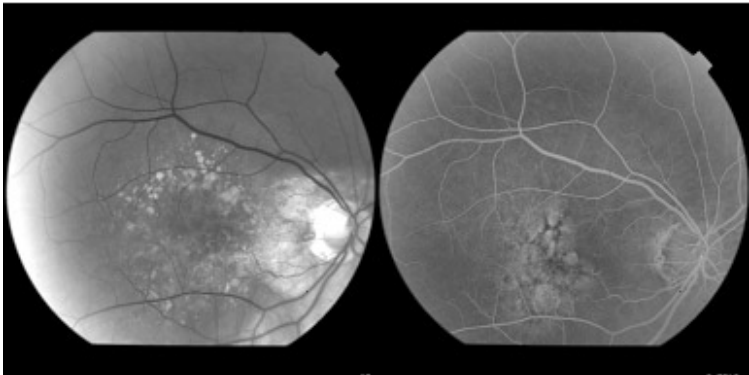
The vasculature in the back of the eye, on the ocular fundus, is a multi-layered structure that supplies the light-sensitive retinal tissue with essential nutrients and assists in the disposal of metabolic waste products [40]. The upper-most layer, referred to hereafter as the “retinal vasculature” is the subject of this chapter.

The retinal vasculature is visible noninvasively using a common clinical instrument known as the fundus camera, or fundus microscope [116]. There are two common types of fundus microscopes: mydriatic and nonmydriatic [61]. Mydriatic instruments are designed assuming that the eyes have been dilated using drops. The nonmydriatic instruments do not require dilation. The mydriatic instruments produce higher-quality images due to the wider field of view enabled by dilation. The nonmydriatic instruments are better suited to initial screening type applications, or in cases when dilation is problematic or impossible. In addition to the above imaging techniques, fluorescence-based angiography methods are in common use as well; fluorescein and indocyanine green (ICG) are the most commonly used dyes. Increasingly, confocal instruments, also known as scanning laser ophthalmoscopes (SLO), are entering clinical use, but are still not as common as the fundus camera [116].

Figure 6.1(a) shows a fundus camera in use. Figure 6.1(b) shows a typical fundus image captured using red-free illumination. The bright round



(a)



(b)

(c)

FIGURE 6.1

Illustrating common retinal imaging methods. Panel (a) shows a fundus camera in use. The lower panels show two images of the same eye containing pathologies. Panel (b) is a retinal image acquired using red-free illumination. The vessels appear darker than the background in this case. Panel (c) was obtained by injecting a patient with fluorescein, and imaging the resulting fluorescence. In this latter image, the vessels appear light against the background. Note in the two panels how the retinal features and pathologies are revealed differently.

structure in this image is known as the optic disk. Retinal arteries and veins are visible as dark linear branched structures. They enter or exit the eye via this location. The darker region in the middle of the image with no vasculature is known as the macula. The center of the macula is largely responsible for acute vision, and is known as the fovea. Many conditions leading to blindness are associated with pathologies of the macular region of the retina. Figure 6.1(c) shows a sample fluorescein angiogram (FA) obtained by injecting a patient with a dye called fluorescein, which fluoresces when excited with light of the appropriate wavelength. The fluorescein travels through the circulatory system and makes the vessels in the retina appear brighter than the surrounding area. Generally, different imaging modalities reveal different layers of the

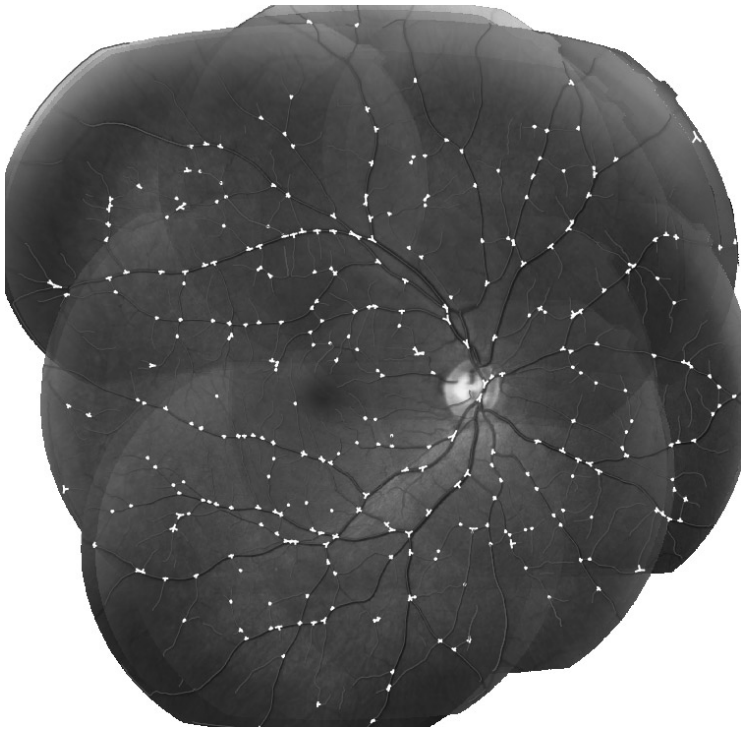


FIGURE 6.2

An automatically generated tracing of the retinal vasculature for a healthy eye, using RPI-Trace, a multi-platform software package developed by the authors. The fundus camera was used to image various portions of the retina, which were montaged (mosaiced) together to form a high-resolution, wide-angle image. The vessel traces are shown in green. The points of bifurcation and crossovers are indicated in white.

overall retinal vasculature at different times. The reader is referred to [40] for a more in-depth description of instrumentation, intra-ocular anatomy, function, and pathologies.

Segmentation and tracing of the retinal vasculature has widespread applicability. Considerable research has been devoted to developing automated algorithms to identify the vasculature structure and differentiate it from other features that may be present in retinal images. Figure 6.2 shows an automatically generated tracing of the retinal vasculature. In this case, the fundus camera only yielded small 30° views of the retina. Multiple such images were combined to form a seamless mosaic (montage) of the retina, which was traced using a software package named RPI-Trace developed by the authors of this chapter.

The quantitative morphology of the retinal vasculature is of direct interest to a number of studies in ophthalmic science. For example, observation

and quantification of the development of retinal vasculature in premature infants is important for early detection of serious conditions such as retinopathy of prematurity [40, 108]. Many diseases related to vision loss are associated with abnormalities in the retinal vasculature. Common examples of such abnormalities are hemorrhages, angiogenesis, increases in vessel tortuosity, and blockages. For this reason, the topological and/or morphological changes in the retinal vasculature are of great interest to clinicians. More generally, the retinal vasculature has the unique distinction of being the only part of the body vasculature that is observable noninvasively. Indeed, the retinal vasculature, which is affected by all factors that affect the body vasculature in general, is a unique “window” into the circulatory system. For this reason, it is studied for nonocular conditions as well, such as hypertension [15, 36, 105, 106]. Finally, the structure of the retinal vasculature is distinctive enough to be used as a biometric for identifying persons. However, this type of application is not widespread due to the high performance of iris scanning technology, which has simpler imaging requirements.

Characteristic points of the retinal vasculature, such as bifurcations and crossover points (see Figure 6.3) are of special interest from the standpoint of developmental studies as well as clinical diagnosis. They form an important component of retinal image understanding systems [1]. For example, arteriolar narrowing, “nicking” or blockages of vessels at crossover points, and retinopathy are associated with hypertension (high blood pressure) [55]. Figure 6.3 (right panel) shows an example of artery/vein nicking.

Changes in the retinal vasculature are important indicators of disease states and/or effects of clinical treatments [48, 74]. Change detection is important for supporting a variety of clinical trials, high-volume reading centers, and for

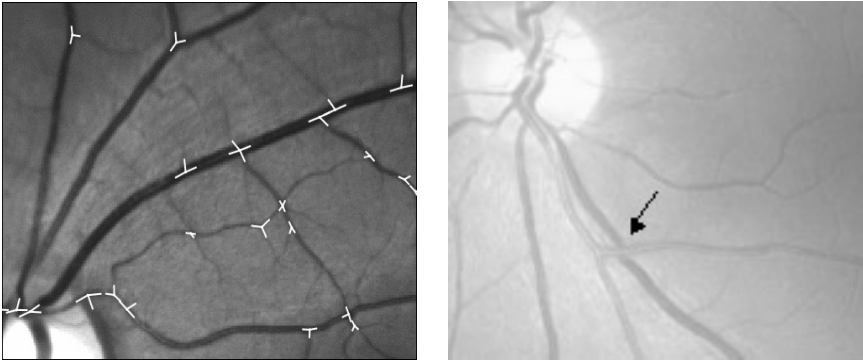


FIGURE 6.3 (See color insert.)

Crossing and branching points of the retinal vasculature. The left panel shows these points as detected by RPI-Trace. The right panel shows an example of vein nicking observed in hypertensive patients caused by an artery exerting pressure on a vein. Automatic location of these interest points may be valuable for clinical diagnosis.

large-scale screening applications. Critical to change detection is the problem of accurate alignment, that is, registration of retinal images that have been captured at different times [11]. Such image capture is performed, for example, during annual checkups. Images are also captured before and after various treatments to observe and document the changes of clinical interest [10, 41]. Finally, image sequences are routinely captured during fluorescein and ICG angiography. The inter-image changes reveal patterns of blood flow in this case. Accurate registration of images ensures that the detected changes (i.e., the difference image) reflects only the real changes, and avoids any artifacts associated with the registration procedure itself. Figure 6.4, panels A and B, shows a pair of images of the same retina taken 30 days apart while the patient was undergoing a clinical treatment. These images have been registered accurately using a sophisticated algorithm developed in this group [22]. The result of detecting the changes are also shown in Figure 6.4C.

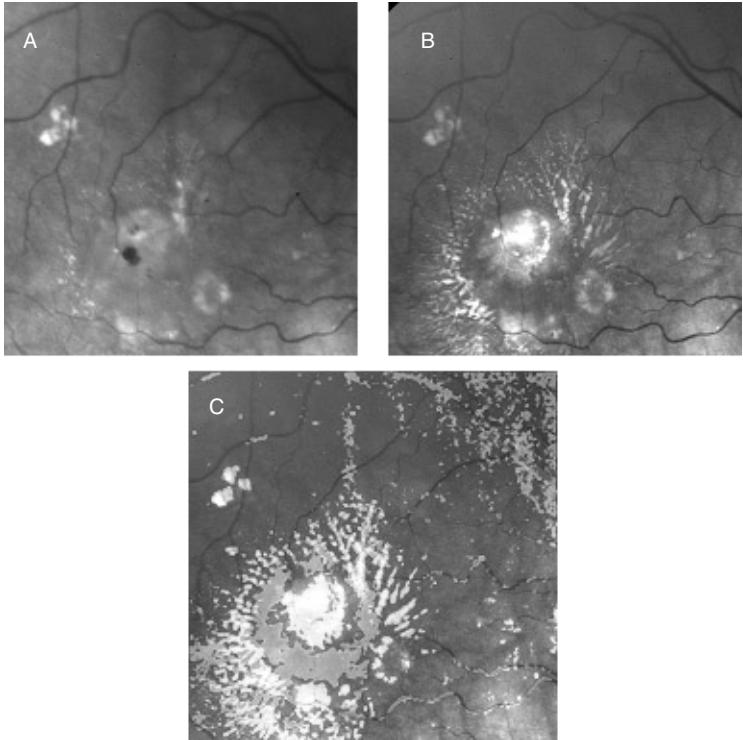


FIGURE 6.4 (See color insert.) Application of vessel tracing and registration to retinal image change detection. Panels A and B show images captured 30 days apart. Panel C shows the result of registration of the above two images, followed by an automatic segmentation (delineation) of the “difference” image, revealing the regions of change in blue.

An essential first step for retinal image registration is precise detection and localization of landmarks, or feature points [17]. These points must have the following properties to be useful. They (1) must be at fixed locations on the patient's retina; (2) must be present in sufficient numbers in all areas of the retina for effective location determination; (3) must be detectable in different images of the same area of the retina even when the images differ in magnification, focus, and lighting; and (4) must be quickly detectable. Points identifying bifurcations and crossing points of the retinal vasculature, and traces of the vasculature between these characteristic points, generally meet these requirements (with exceptions such as those arising in retinal detachment). That is, the retina appears as a rigid structure unless detached. Physicians have long used vasculature features as spatial references for finding specific locations on the retina during manual examination and surgery. [Figure 6.1](#) shows an automatic tracing of the retinal vasculature with the crossing and branch points highlighted. Section 6.8 describes methods for locating these points with sub-pixel precision and high repeatability.

Image registration has direct applications in addition to change detection [27, 28, 84]. One such application is mosaic synthesis, which requires multiple partial views of the retina to be combined into a single high-resolution synthetic wide-angle view [69, 70]. Retinal mosaics are useful for accurate vessel morphometry taking overlapping views into account, detection of changes at the retinal periphery for conditions such as AIDS CMV [128]. Mosaics are also useful as key elements in building a spatial map [9, 14, 95] of the retina for the purpose of tracking and real-time spatial referencing. [Figure 6.5](#) illustrates retinal image mosaicing and spatial referencing as registration problems. Diagnostic images (panels A through E) are mosaiced together to form a wide-angle spatial map of the entire retina [18–20, 23].

The retinal vasculature has been used for real-time tracking [5, 6] and spatial referencing [95] in the context of laser retinal surgery. Tracking algorithms use the retinal vasculature to register each frame of an image sequence to the next. However, given the tendency of the retina to exhibit large motions [88], tracking is often lost or inaccurate. Whenever this happens, tracking must be reestablished [8, 9]. One method to reestablish tracking is to use the mosaic as a spatial system of reference, as described by Becker et al. [10]. Another approach is to register each image frame to a preestablished mosaic-based map of the retina. This is known as spatial referencing. Shen et al. [95] have described a fast method for spatial referencing using quasi-invariant indexing. In this method, geometric invariants are computed from local constellations of landmarks (crossing and branching points of the retinal vasculature), and looked up in a precomputed indexing database to estimate an initial registration and iterative alignment of the vasculature. This initial registration is subsequently refined using robust estimators. The spatial map consists of mosaics, together with the surgical plans (outlined in panel F) drawn by the physician, and various data structures (quasi-invariant indexing databases) to support fast spatial referencing during surgery.

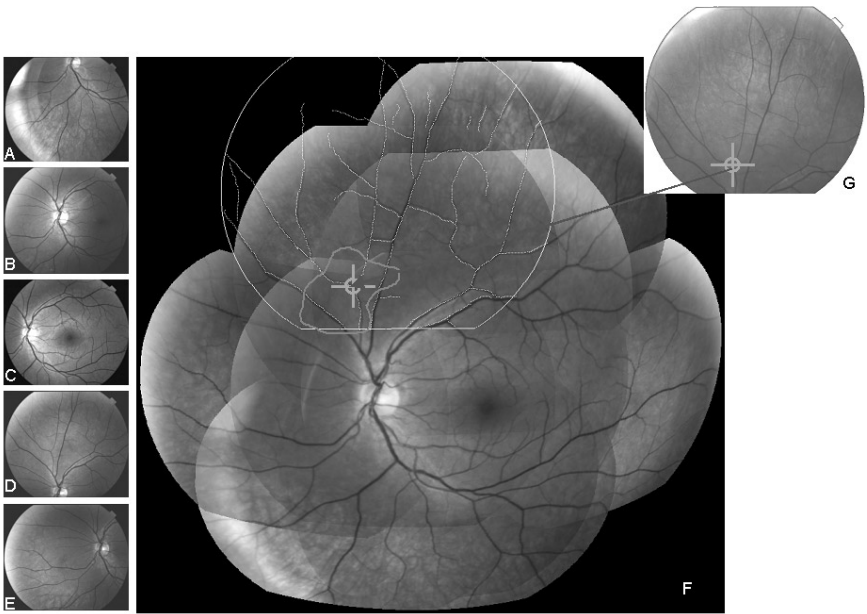


FIGURE 6.5

Two applications of vessel segmentation: mosaicing and spatial referencing. Panels A through E show some of the images taken from the same eye on a single visit. Panel F shows the mosaic (or spatial map) built from these and other images [23] using traces of the vasculature as spatial landmarks, and with a desired treatment area outlined. Panel G simulates an image acquired during a surgical procedure in which the aiming point in G is referenced against the spatial map in real-time to determine the precise location of the laser and check if it is within the desired treatment area before allowing the laser to be turned on [95].

Figure 6.5 illustrates spatial mapping. Panel G simulates an image frame captured by a camera during a surgical procedure. This image is registered onto the spatial map, effectively estimating the location of a laser beam (illustrated by the cross-hairs on panel G) relative to the spatial map. The arrow illustrates this mapping. The vascular traces from the image frame (panel G) are superimposed on the mosaic to show correct registration. Spatial referencing can, for example, determine whether or not the surgical laser is aimed within a desired treatment region. While performing laser retinal surgery, the images that the surgeon is viewing can be processed for vascular landmarks in much the same fashion as they were extracted for registration. These landmarks can then be compared against the landmarks in the mosaic image so as to identify whether the surgical tool is in the correct region. If so, the surgeon can then apply the laser to the desired treatment area, while the spatial referencing algorithms enable a recording of the optical dosage administered to each point. If not, the spatial referencing processor can perform a protective cut-off of the laser.

Thus, the extraction of vessels is vital in providing the landmarks necessary to perform such spatial tasks in computer-assisted surgical instrumentation.

6.3 Retinal Imaging Background

Still digital images and live digital image sequences of the human retina are captured using a digital video camera attached to a fundus camera [42, 85]. Still photography is generally performed with a flash and live imaging is done with continuous illumination. In the former case, imaging noise is lower than in the latter case where less light is available. The confocal imaging case is not considered here [116]. Readers who do not have access to a source of retinal images, such as an ophthalmology clinic, can access large collections of retinal images on CD-ROM [123].

Several aspects of retinal images make automated processing challenging. First, the images are highly variable. Large variability is observed between images from different patients, even if healthy, with the situation worsening when pathologies exist. For the same patient, variability is observed under differing imaging conditions and during the course of treatment. Unlike industrial vision problems where the conditions can be carefully controlled, retinal images are frequently subject to improper illumination, glare, fadeout, and loss of focus. Artifacts can arise from reflection, refraction, and dispersion.

The movement of the eye can result in significant variation, even for a stable, well-seated patient. The eye movements are rapid and significant [88]. The saccadic movements involve sudden jumps up to 15° . These movements occur at speeds ranging from 90 to $180^\circ/\text{sec}$. The mean peak acceleration can range from 15,000 to $22,000^\circ/\text{sec}^2$. These movements cannot be fully suppressed, even with medication. Often, medications are considered harmful, and so are not used. These numbers imply that interlaced image sensors are often a poor choice for imaging the vasculature [4]. Most commercial imaging systems, such as the TOPCON ImageNet system, employ non-interlaced digital megapixel cameras. Additionally, movements resulting from a distracted patient, or in response to the irritation induced by the laser, or the involuntary attempt to fixate the fovea on the laser light, all affect the quality of the image. Finally, each image or frame in a video sequence represents a partial view of the retina, and the zoom factor (magnification, scale) can vary from frame to frame, due to either selection of a different magnification setting on the fundus camera or movement of the camera nearer to or farther from the patient's eye (which may be necessary for focusing).

The inherent difficulty of illuminating the retina in a uniform and steady manner for reflectance imaging [100] causes several difficulties. For effective imaging, the retina cannot simply be illuminated by directing a strong light

into the patient's eye. In addition to causing patient discomfort, this would generate reflections off the patient's cornea (glare) that can dominate the light reflected from the retina back out the pupil. As a result, another method of retinal illumination is used. Before imaging a patient's retina, the patient is administered eye dilating drops (e.g., Tropicamide ophthalmic solution) to dilate the pupil to allow a large area for light to enter and exit the eye. The fundus camera's illumination is focused to form a ring of light at the anterior (front) of the eye with an unilluminated circle in the center of the pupil. As a result, reflection off the surface of the eye only occurs in the illuminated ring. At the retina, the light ring is out of focus, and the result is diffuse illumination. The aperture of the fundus camera is designed to block the reflected light from the circle of illumination while allowing light from within the unilluminated circle, resulting in a fundus image uncontaminated by reflection or glare from the cornea. Unfortunately, this method does not always work flawlessly. For the retina to be illuminated while preventing glare, the camera must be focused properly on the patient's eye. Movements of the patient's eye, even slight ones, can degrade the focusing. In particular, if the eye moves so that the iris blocks some of the ring of illumination, the total illumination of the retina can drop dramatically. In addition, eye movements can allow some stray reflected or refracted light to be directed into the aperture of the fundus camera. This is seen as a bright glare at the edges of the retinal image. Figure 6.6 shows examples of these possibilities.

When live video imaging is desired, one has to contend with eye movements and the difficulty of providing steady illumination. The factors noted above result in image frames that are often dim, out of focus, motion blurred, or corrupted with optical effects such as glare or nonuniform illumination. In retinal still photography, skilled technicians quickly refocus the camera for optimum illumination for each picture. Images are only taken/saved when the illumination is optimal. Any images with insufficient illumination or excessive glare are simply discarded. Tracing algorithms designed for real-time

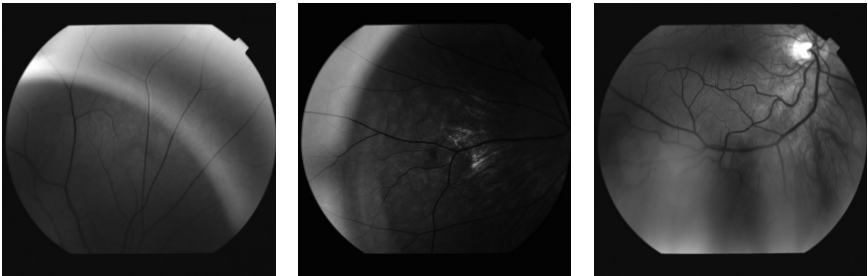


FIGURE 6.6

Some examples of fundus images with nonuniform illumination. Note in the rightmost image how the lower portion of the image is blurred.

online operation do not have this luxury. They must be able to work with image frames with suboptimal illumination, and be able to detect when image quality is too poor for processing, and then reject these frames. They must not only be adaptive enough to be useful and reliable, but also efficient enough to handle high data rates. While much work remains to be done in this area, much progress has been made. Some of the algorithms described in Section 6.5 were designed in the context of such conflicting, practical, and extreme needs.

6.4 Models for Detecting Retinal Vasculature in Digital Imagery

This section describes methods for modeling the appearance of retinal vessels in fundus images in a manner that is sufficient for the purpose of image segmentation. Specifically, we are interested in features of vessels that allow an algorithm to determine if specific pixels in an image are part of the vessel tree or the background. A full physics-based modeling of the image formation process, including the geometry of the vasculature and the imaging systems, could yield more accurate and realistic models, but is beyond the scope of this chapter (see [16, 35, 89, 100]).

Generally, models are applicable only in the context of the image type for which they are developed. For example, in a red-free retinal image (that is, an image taken with a specialized red filter to block out most of the red light), the blood vessels appear to be darker than the local background. In a fluorescein image, dye is injected into the bloodstream, which causes the vessels to appear brighter than the background. Application of the same intensity-based model in both cases will not yield the same results. Fortunately, it is often possible to modify a model that works well at detecting light vessels against a dark background to be able to detect dark vessels against a light background.

Most of the methods described in the literature model the cross-sectional profile of vessels. Figure 6.7 illustrates the idea of a cross section. The conceptual basis for such modeling is the differential behavior of light propagation through vessels, and the reflectance off of the vessel surface, compared to the local background. The following sections review some of the different models used to detect retinal vessels and assume an image in which blood vessels are darker than the surrounding background.

6.4.1 Thresholding

One approach to segmentation of blood vessels is based on the observation that blood vessel pixels are darker than the background pixels. Thus, by identifying all pixels that are darker than some threshold T , you can easily identify

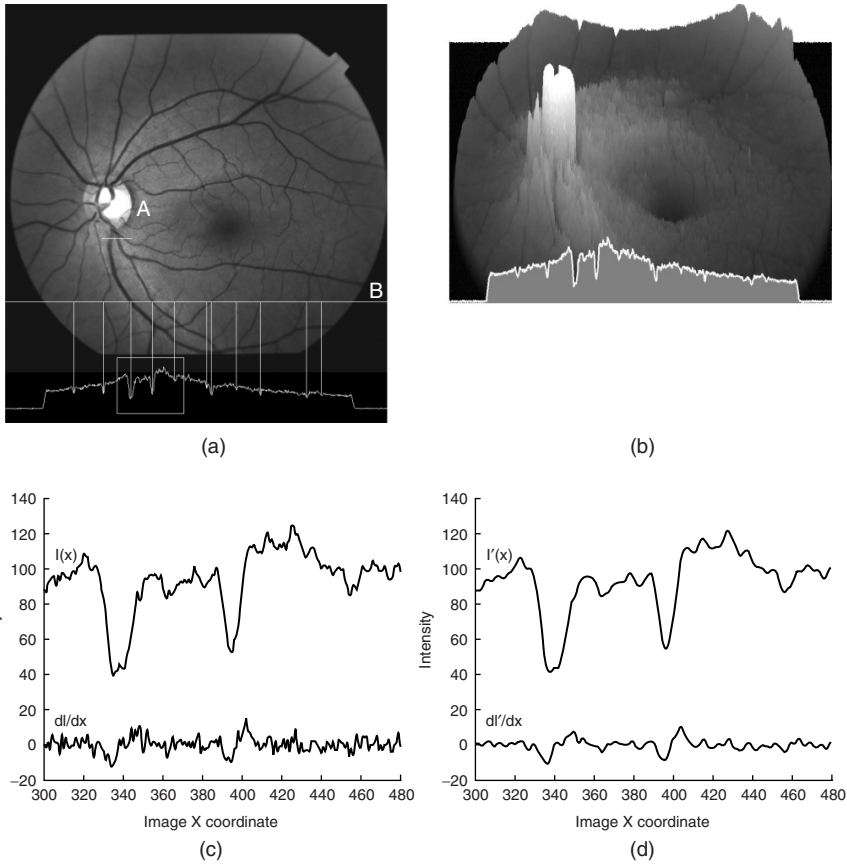


FIGURE 6.7

Cross-sectional gray-scale profiles of retinal vasculature. Panel (a) shows an image with two marked cross sections, A and B, with the intensity profile of B below the image. Lines are extended from the points where vessels intersect the cross section to the profile below showing how vessels correspond to relative minima on the profile. Panel (b) shows a three-dimensional image where the pixel intensity is used as the value for height and illustrates the resulting intensity profile from a cut along the cross-section B. Panels (c) and (d) show the intensity and derivatives for the actual and smoothed profiles, respectively, for the boxed region shown in panel (a).

all vessel pixels. While attractive in its simplicity, this idea does not generate satisfactory results. First, a method for selecting an appropriate value for the threshold needs to be addressed. Second, retinal images can contain pathologies, are inherently noisy, and the intensity structure is not uniform throughout the entire image. Figure 6.8 exemplifies how the intensity structure varies through an image. So, a threshold that works in one area of an image does not work in another or incorrectly identifies noisy pixels from pathologies or the

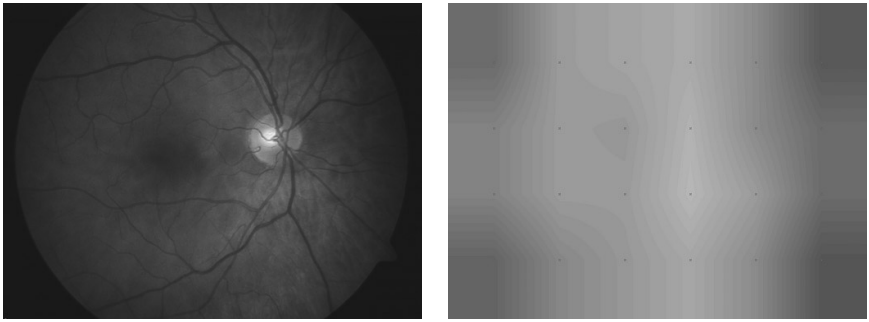


FIGURE 6.8 (See color insert.)

Illustrating the changes in local thresholds indicative of the varying intensity structure across the entire image. For each region center denoted by the x in the right image, a local minimum threshold is calculated. These thresholds are then interpolated for any pixel that is not a region center. The thresholds shown in the image on the right were determined from the image on the left with the thresholds multiplied by 4 for display purposes.

background. Thus, thresholding by itself is not an adequate model of vessel segmentation.

6.4.2 Local Gray-Scale Minima-Based Models

A different way to model the vasculature is from a slight modification to the above dark-pixel observation in that the central portions of vessels correspond to maximum absorption of light and are the *darkest* pixels [29]. Thus, a method of finding the darkest pixels relative to their neighbors should indicate that the pixel is a vessel. The easiest way to find these minima is to consider cross sections, commonly in both the x - and y -directions. By scanning a row or column of pixels and flagging each pixel that is a relative minimum (i.e., its intensity is smaller than its two neighbors), a vessel pixel can be found.

While the simplicity of this model is attractive, it has limitations. First, it is possible to have flat portions of the cross-sectional profile, including in the vessel. Second, the local minima are not unique to vessels — the image background often contains relative minima caused by pathologies, the presence or absence of pigmentation in the retinal tissue, noise from image signal path (through the lens, through the vitreous fluid, etc.), or noise from the camera. Thus, while this model is simple and relatively fast, it often flags as many non-vessel pixels (false positives) as it does vessel pixels. Nevertheless, this model is proven to be useful as a way of obtaining “seed” or “candidate” vessel points from which to initiate a vessel tracing algorithm if coupled with a method to cull out bad points [21]. [Figure 6.9](#) shows a sample result of the local minima model.

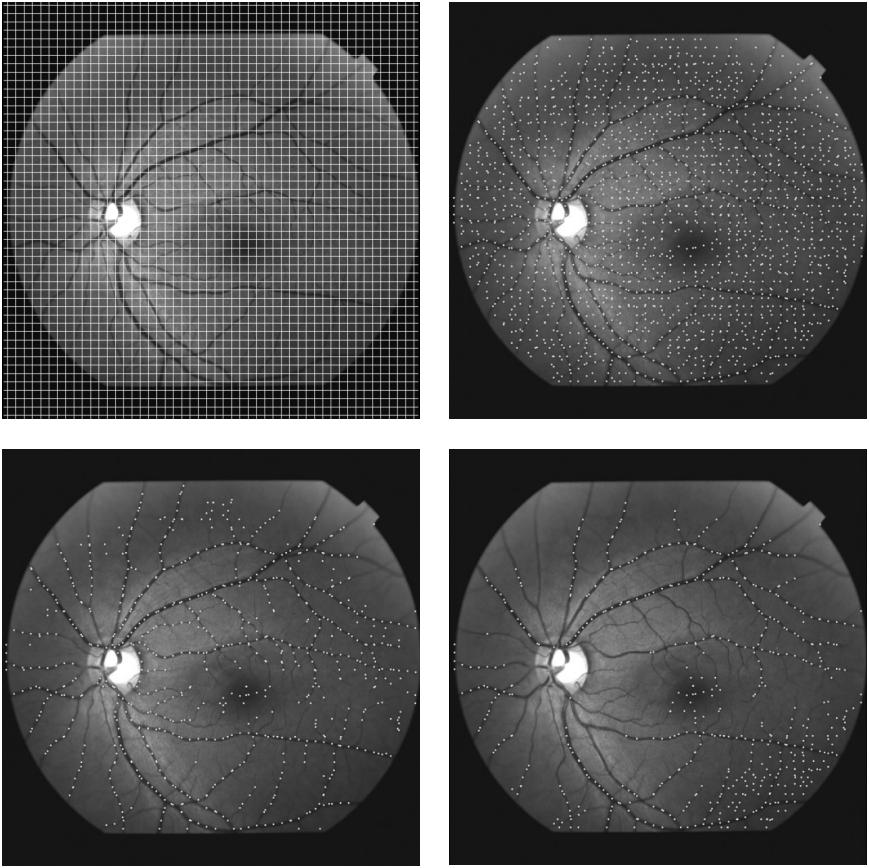


FIGURE 6.9

The performance of three different models detecting retinal vasculature. Panel (a) shows the 50 horizontal and 50 vertical lines used as cross sections to extract profiles to identify vessels. Panel (b) shows local minima only. Panels (c) and (d) show local minima with local and global thresholds, respectively, for $\alpha = 1$ in Equation 6.1.

6.4.3 Combining Local Gray-Scale Minima with Thresholding

The above model can be enhanced by combination with thresholding. From the observation that the intensities (pixel values) of the relative minima in the background region of the image are substantially higher than the local minima found in vessels, a threshold can be selected such that if the relative minimum is higher than the threshold, then that pixel will not be considered as part of a vessel. Only gray-scale local minima with intensities below the threshold should be considered vessel pixels. The trick is in determining an appropriate threshold.

One method is to use a single global threshold value for the entire image. A straightforward method for estimating such a threshold is to examine the intensities of a set of evenly spaced sample points S throughout the image and to use order statistics to arrive at a threshold. First, the intensities of the sample points denoted S are ordered. Then, the median $m(S)$ of these values is computed. Additionally, the median of the absolute deviations of all the points from the median value is then computed and converted to an estimate of the scale and denoted as $s(S)$. The threshold $T_m(S)$ can then be defined as:

$$T_m = m(S) - \alpha s(S) \quad (6.1)$$

where α is a non-negative weighting factor. A typical value for α is 1.

A somewhat more adaptive approach is to estimate the above thresholds locally. In this approach, the image is tiled into rectangular regions (which may overlap), and the threshold is calculated as above for each such region. For determining T_m for points in between the centers of these rectangular regions, bilinear interpolation can be used. The advantage of using local thresholds is that they are more appropriate due to the change in contrast and lighting conditions observed in a single image but they are computationally more expensive. Results of filtering image pixels over a 50×50 grid using local minima with global and local thresholds computed according to Equation 6.1 are presented in Figure 6.9. The variation in the local threshold across an image is displayed in Figure 6.8.

6.4.4 Hybrid Gray-Scale and Edge-Based Models

As noted above, a purely gray-scale (intensity) based model is inadequate for separating vessel and non-vessel pixels. Even after combined with thresholding as described above, local gray-scale minima are incorrectly identified as vessel pixels. Additional information is needed to perform a more complete separation. Vessel boundaries provide powerful *structural* information for this purpose. All pixels in a blood vessel are bounded on two sides by a pair of prominent boundaries. Applying this idea, for each gray-scale minima detected, areas on either side of the minima can be searched to find edges by looking for oppositely signed derivatives where the derivative at each pixel is computed using the following equation (for a horizontal profile):

$$I'(x, y) = I(x + 2, y) + 2I(x + 1, y) - 2I(x - 1, y) - I(x - 2, y). \quad (6.2)$$

When the pixel is part of the background or inside a vessel (depending on the vessel width), the derivative should be very close to zero. When a pixel is on the left or right vessel boundary, the derivative magnitude is at its maximum and is negative for the left boundary and positive for the right boundary. Only derivatives with the greatest local magnitude (i.e., local minima or local maxima) are considered as boundary points.

The above edge constraints serve to filter out local gray-scale minima that are not associated with boundary edges. This constraint by itself is not sufficient to distinguish vessel and non-vessel pixels. This can be seen by noting that a pair of edges on opposite sides of the image are obviously too far apart to form a vessel that satisfies the above requirements. Therefore, it is necessary to filter the points based on the distance between the edges. If the oppositely signed derivatives are too far apart to be representative of expected vessel widths, they should not be considered matching boundary points of a vessel and the local minima should be considered background.

This type of bound on expected vessel widths is a powerful constraint. The results of using this method to filter image pixels using 50 evenly spaced vertical and 50 evenly spaced horizontal cross sections (50×50 grid) can be seen in Figure 6.10. This method improves upon the results shown in Figure 6.8. This is the method used by Shen et al. in their real-time vessel tracing work [96].

Even with a bound on vessel width, the relative minima with edges model is often inaccurate in identifying vessel pixels. Specifically, it is still common to incorrectly label background pixels as vessel by encountering multiple oppositely signed derivatives around a minima within a given distance in the background regions. One method to be more selective is to check if the magnitudes of the oppositely signed derivatives are roughly equal. While this is effective in eliminating some of the false positives, there is still one more step that can be applied to eliminate even more false positives. This involves using a thresholding technique similar to that described in Equation 6.1 above, but applied to the edges and based on the edge strength.

6.4.5 Combining Hybrid Gray-Scale and Edge-Based Models with Thresholds

As can be seen in the results in Figure 6.10, the hybrid gray-scale and edge-based model still incorrectly identifies non-vessel pixels as vessel. One approach to minimizing such false detection of vessel pixels is to establish thresholds based on edge strengths. This threshold would represent the cutoff between what should be considered a vessel boundary and what should just be considered background noise. This threshold can be calculated globally or for local regions using the procedures described earlier. For each sample in a given region R (either a global or local region), the derivatives in four directions, 0° , 45° , 90° , 135° , are calculated. The mean and standard deviation of these derivatives are descriptive of the contrast for neighboring pixels or potential edge sites within the region and can be used to develop a minimum edge strength threshold, T_e , for an edge within a region R as follows:

$$T_e(R) = \mu_d(R) + \alpha\sigma_d(R), \quad (6.3)$$

where $\mu_d(R)$ is the mean of the all the derivatives in the region, $\sigma_d(R)$ is the standard deviation of the derivatives in the region, and α is some weighting factor applied to the standard deviation. A typical value for α is 1.5.

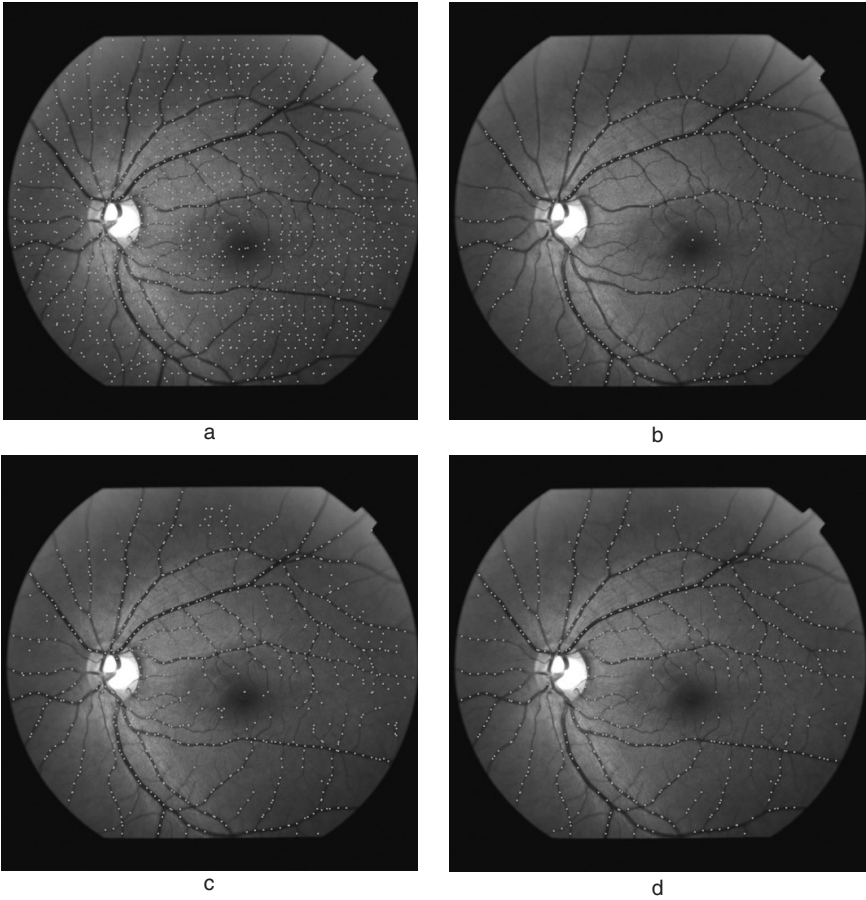


FIGURE 6.10 (See color insert.)

The performance of edge-based vessel models. Panel (a) shows the identified vessels from the vertical and horizontal profiles shown in [Figure 6.9\(a\)](#) using one-dimensional parallel edges within a radius of 11 pixels and edge magnitude within 20%. Panel (b) shows use one-dimensional parallel edges with the same settings as used in panel (a) with the addition of global thresholds for grayscale minima and edge strength. Panel (c) is the same, but with local thresholds. Both (b) and (c) used $\alpha = 1.5$ in Equation 6.3 and $\alpha = 1$ in Equation 6.1. Panel (d) shows vessel points found using two-dimensional parallel edges (with the same local thresholds as panel (c)).

Anything above this threshold would be considered sufficiently strong to be considered an edge.

This threshold, coupled with the gray-scale minima threshold discussed previously, now provides a method for identifying a vessel point in a more selective manner. Thus, for every minima below some threshold, if it has two, roughly equal, oppositely signed derivatives within a defined distance, where the left derivative is negative and the right is positive, if the sum of

the magnitudes are greater than $2T_e(R)$, one could consider those points as vessel boundaries. Results from using this model are shown in [Figure 6.10](#).

6.4.6 Models Based on Parallel and Opposite Two-Dimensional Edges

Up until now, our discussions of vessel models have been based entirely on a one-pixel-wide cross-sectional profile. However, there are other, non-one-dimensional properties of vessels that can be used to improve the accuracy of our models. One such property is the observation that while vessels do curve, they are locally straight over short distances. This property allows us to expand the one-dimensional models described above into two-dimensional models.

Consider a vessel that is locally oriented along the vertical (i.e., y) axis over a distance of L pixels. For such a vessel, the right boundary (edge) can be detected using a kernel of the following form:

$$Length = L = \left\{ \begin{array}{|c|c|c|c|c|} \hline -1 & -2 & 0 & 2 & 1 \\ \hline -1 & -2 & 0 & 2 & 1 \\ \hline & & \vdots & & \\ \hline -1 & -2 & 0 & 2 & 1 \\ \hline \end{array} \right.$$

This is the operator used by Can et al. [21], and called “low-pass differentiators” by Sun et al. [110]. The kernel response will be a maximum positive number when the left half of the above kernel is overlaid on a dark vessel and the right half is overlaid on the brighter background. Thus, it yields a maximum response when the column of zeros is on the right boundary of a vertical vessel. A mirror image about the vertical axis of the above kernel has similar response properties, but for the left boundary of the vessel.

The kernel length L is set based on two considerations. First, a higher value of L results in a more sensitive detection of the two-dimensional vessel boundary. Specifically, if the gray-scale contrast between the foreground and background pixels is δ , then the kernel response is simply $3 \times L \times \delta$. This suggests maximizing the length of the kernel L . However, this cannot be done indefinitely, given the curvature of vessels. Another consideration that is important for high-speed tracing applications is the computation cost. Increasing L results in a proportional increase in the computation time. A reasonable trade-off is to fix L for a given application, or a class of images. For example, Can et al. [21] used a value of 6. As an aside, the computations for this kernel are quite efficient. The constants are powers of 2, so it only involves fixed-point shift and add operations.

Vessel boundaries at arbitrary angles can be detected by kernels that are obtained by rotating the kernel in Equation 6.2. Fortunately, a quite coarse discretization of angles is known to be adequate for retinal vasculature based on its known curvature properties. For example, Can et al. [21] used a set

of 16 kernels separated by 22.5° for orientations from 0 to 360° , while Chaudhuri et al. [25] used just 12 kernels separated by 15° for orientations from 0 to 180° .

Once the templates for finding vessel boundaries have been created, the task of locating vessel pixels is similar to that described in Section 6.4.4 above. For a pixel to belong to a vessel, there must be edges around a local minima (as indicated by two template responses within a given distance in opposite directions) that are nearly equal. The thresholding criteria described above (T_m and T_e) are also applicable, except that prior to applying the T_e threshold, the response must first be normalized to account for the length. Vessel points identified by these criteria can be seen in Figure 6.10.

6.4.7 Models Based on Gaussian Cross-Sectional Profiles

Another methodology used to model a cross section of a blood vessel is fitting the observed cross-section profile of the vessel with the profile of a predefined curve. Gaussian profiles have been shown to be effective in this regard [25, 26, 127]. For a vessel that is locally oriented along the y -axis (i.e., vertically), and assuming a Gaussian gray-scale profile yields the following model:

$$G(x) = b - c \times e^{-(x-a)^2/2\sigma^2}$$
 (6.4)

where a is the center point of the vessel, σ is a parameter representing the width of the vessel, b is the approximate value of the background around the vessel, and c represents the degree of contrast between the vessel and the background (i.e., how dark the vessel is as compared to the background). It can be observed from the images in Figure 6.11 that while not entirely accurate

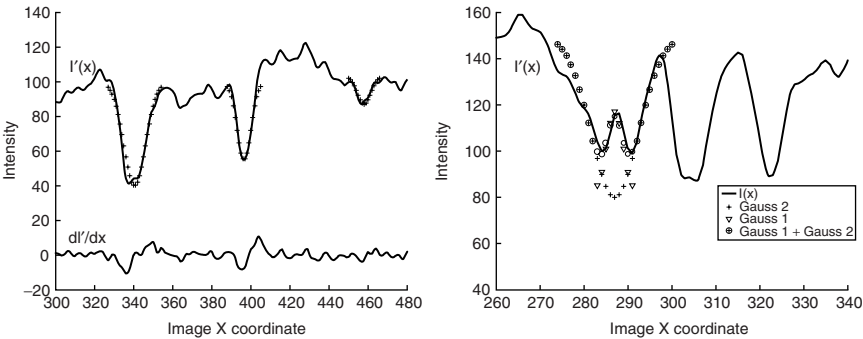


FIGURE 6.11 Illustrating the use of a Gaussian curve to model the gray-scale vessel cross section. The left panel shows how three different Gaussian curves formed using Equation 6.4 with differing values for σ closely approximate the smoothed profile “B” from Figure 6.7. The right panel shows how for a vessel with central reflex obtained from the profile marked “A” in Figure 6.7, two Gaussian curves can be combined to better approximate the vessel profile.

for the entire profile, the Gaussian curve does provide a close fit to portions of a cross section corresponding to the vessel.

The Gaussian model described above can be used to construct matched filters [25]. Constructing a kernel based on the Gaussian profile is challenging. First, because the function in Equation 6.4 is defined for all points (i.e., it is continuous), the spatial extent of the filter must be made finite to enable efficient computation. Commonly, a filter formed from Equation 6.4 is truncated beyond two or three standard deviations; that is, $a - 3\sigma \leq x \leq a + 3\sigma$.

Also, it is convenient to design the kernel such that the template response is zero in a uniform region like background and non-zero over vessels. One such kernel, used by Chaudhuri et al. [25], used a kernel of the following form based on a value for σ of 2:

4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4
---	---	---	---	----	----	----	----	----	---	---	---	---

The maximum response results when the central entry (i.e., -6) coincides with the center of the vessel. Thus, this template can be applied at all pixels across a cross section, and the point at which the response is maximum can be considered a center point of a vessel.

This filter described above can now be expanded into a two-dimensional model to once again take advantage of the locally straight feature of vessels. This is accomplished in the same manner as described in Section 6.4.6. Once again it becomes necessary to generate a set of templates to account for different orientations of vessels. A sample two-dimensional kernel for a Gaussian profile and a kernel length L is shown below.

$$Length = L = \begin{cases} \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 4 & 3 & 2 & 1 & -2 & -5 & -6 & -5 & -2 & 1 & 2 & 3 & 4 \\ \hline 4 & 3 & 2 & 1 & -2 & -5 & -6 & -5 & -2 & 1 & 2 & 3 & 4 \\ \hline & & & & & & \vdots & & & & & & \\ \hline 4 & 3 & 2 & 1 & -2 & -5 & -6 & -5 & -2 & 1 & 2 & 3 & 4 \\ \hline \end{array} \\ \end{cases}$$

The above model requires an estimate of the scale σ . In the implementation described by Chaudhuri et al. [25], a fixed value of σ was used and determined to be effective even for detecting vessels of varying scales. Figure 6.13 illustrates the effectiveness of an algorithm using this model σ .

Sato et al. [91] propose using multiple Gaussian filters using a small number of σ values (typically three or four different values are used) to account for vessels of different widths. Gang et al. [44] have proposed a method that relies on adaptively estimating the σ value. They point out that the Gaussian filters, while appropriate as vessel models, do not directly permit measurement of the vessel width. They propose an amplitude modified second-order differential Gaussian model of the form:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma^3}(x^2 - \sigma^2)e^{-x^2/2\sigma^2} \tag{6.5}$$

With this model, there is a linear relationship between the parameter σ and the diameter of the vessel. The notion of vessel diameter used in this model is distinct from one that can be measured by locating the vessel boundaries.

6.4.8 Models for Vessels with a “Hollow” Appearance

Frequently in fundus images, vessels sometimes appear to be “hollow,” that is, the walls of the vessels appear as dark pixels and the center of the vessels appear lighter, sometimes as light as the background itself. This is caused by a reflection from or below the surface of the vessel and is known as vessel reflex. Figure 6.12 illustrates this phenomenon and more information can be found in [89, 45]. While some of the previously mentioned models are still successful in identifying these vessels [21], none of the models were developed with this phenomenon specifically in mind. To accurately model these “hollow” vessels, a twin-Gaussian model has been proposed by Gao et al. [45].

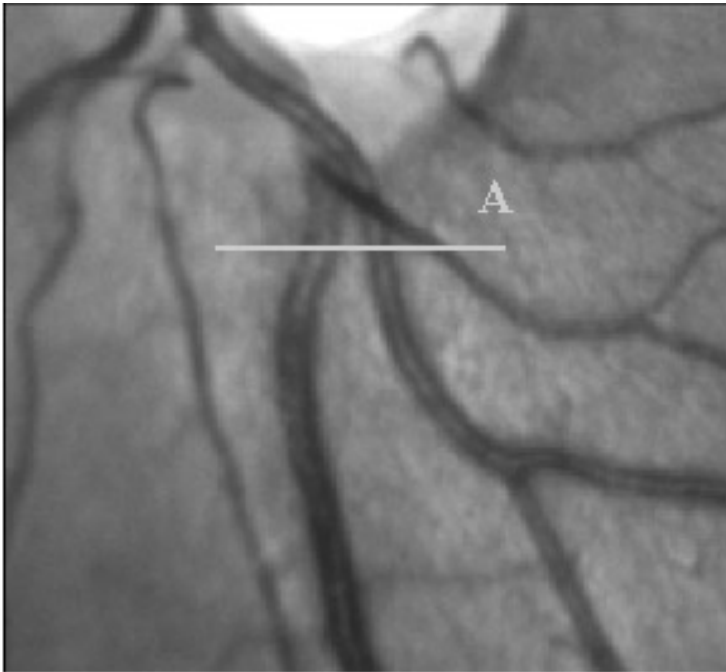


FIGURE 6.12

Magnified portion of the image in Figure 6.7 depicting the central reflex phenomenon, that is, vessels that appear lighter in color in the center, as if they were hollow. This is caused by the reflectance of light on or below the surface of the vessel. See [16, 45, 89] for a more detailed explanation.

The first Gaussian in this model is representative of the entire vessel, whereas the second Gaussian represents the reflex and they are combined as follows:

$$\begin{aligned} C_{model}(x) &= C_{vessel}(x) + C_{reflex}(x) \\ C_{vessel}(x) &= b - c \times e^{-(x-a)^2/2\sigma_1^2} \\ C_{reflex}(x) &= d \times e^{-(x-\epsilon)^2/2\sigma_2^2} \end{aligned} \quad (6.6)$$

The parameters a , b , and c are as described above in Equation 6.6; d is representative of the relative contrast between the vessel pixels and the reflex pixels, ϵ is the center point of the reflex, and σ_1 and σ_2 are representative of the widths of the vessel and the reflex, respectively. This can be seen in Figure 6.11b. Given this equation for a cross section of a vessel with a central reflex, a filter can be generated for the one-dimensional case and for the two-dimensional case in exactly the same manner as described in the previous section and used to detect the center points of these types of vessels.

This section has summarized some of the models used for retinal vasculature that have been used for vessel segmentation. None of the models is individually fully adequate for all applications and the models presented are not all-inclusive. For example, Pedersen et al. [83] have described a vessel model based on using a six degrees-of-freedom cubic spline and its derivatives to locate the boundaries of vessels. This work is not described here. Overall, a combination of these modeling ideas is considered the best approach to vessel segmentation. Of course, the more sophisticated models are associated with higher computational costs, making them less attractive for real-time applications. With rapid improvements in computing speeds, applications can be expected to incorporate increasingly more sophisticated models. In our experience, the kernels based on low-pass differentiators [21, 96] have proved an excellent trade-off when computation speed is important and provide excellent results. When computation times are not critical, a combination of preprocessing algorithms using the Gaussian model and algorithms based on the low-pass differentiators could also be considered.

6.5 Vessel Extraction Approaches

The models described in the previous section lead to a variety of approaches for conducting segmentation and tracing. In this chapter, the term “vessel segmentation” is used to imply the process of classifying each pixel in the image as either belonging to a vessel or not. The term “vessel tracing” is the closely related but refers to the distinct process of extracting the centerlines of the vessels.

There are two prevalent approaches for extracting vascular structure from retinal images. The first approach requires certain processing to be done for

each pixel in the image and will be referred to as pixel processing algorithms. The second approach is known as vectorization or exploratory tracing. These exploratory methods start at an initial point that is determined to be part of a vessel and then “track” that vessel through the image as far as possible. In general, these methods are computationally much faster than pixel processing approaches as only the pixels in the immediate vicinity of the vessel structure are actually “explored” while extracting the vasculature structure. This is the reason they are referred to as “exploratory” algorithms. Both pixel processing and exploratory algorithms are discussed in the following sections.

6.5.1 Pixel Processing-Based Algorithms

The pixel processing-based approaches [49, 53, 86] work by adaptive filtering, morphological preprocessing segmentation, followed by thinning and branch point analysis. These methods require the processing of every image pixel and multiple operations per pixel. When these operations are highly regular, they can be implemented on fast workstations [86] and pipelined accelerators [25]. Other pixel processing approaches involve the use of neural networks [58, 99] and frequency analysis [111] to determine if individual pixels are vessels. Generally, the computational needs of pixel processing methods scale sharply with image size, and are usually unsuitable for fast, real-time processing without special hardware.

The method of Chaudhuri et al. [25] is based on two-dimensional matched filters. These filters are designed to maximize response over vessels and minimize the response in background regions. Prior to matched filtering, the images are smoothed using a 5×5 median filter to reduce the effect of spurious noise. This algorithm uses a Gaussian vessel model similar to that described earlier in Section 6.4. In this case, a vessel profile is described by the Gaussian function

$$f(x, y) = A(1 - ke^{-d^2/2\sigma^2})$$

where d is the distance between the point and the center of the vessel, σ is representative of the width of the profile, A is the intensity of the local background, and k is a measure of the contrast between the vessel and the local background. From this profile, a kernel K is developed that is defined by:

$$K(x, y) = -e^{-x^2/2\sigma^2} \quad \text{for } |y| \leq L/2 \quad (6.7)$$

where L is the length of the kernel, which is the parameter associated with the length of a vessel. This algorithm bounds x at $\pm 3\sigma$. The kernel given in the above equation matches (i.e., yields a maximum response for) a vertical vessel that is locally straight. For different orientations, a kernel with 15° of angular resolution is constructed by simply rotating it accordingly. The value of L is set to 9, which is experimentally determined to work well with both “normal” and highly tortuous (curvy) vessels. The value of σ is fixed and set to 2.

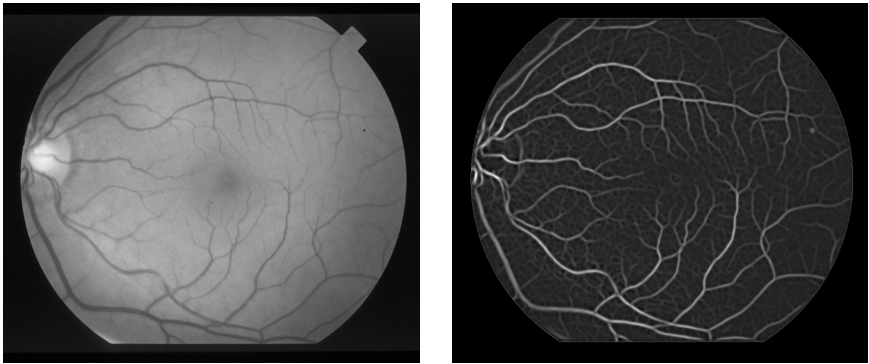


FIGURE 6.13

The use of a matched filter for enhancing vessel-like structures. The left panel shows a retinal image. The panel on the right is the result of matched filtering using the algorithm of Chaudhuri et al. [25].

All the directional kernels are applied at every pixel in an image, with the maximum magnitude response being retained as the value for that pixel in the results image. The results image obtained from this operation can be seen in Figure 6.13. If this maximum is above a certain threshold, the pixel is labeled as vessel. This threshold is determined by selecting the value that maximized the inter-class intensity variance as described in [81].

6.5.1.1 Morphological Filtering-Based Algorithms

The next class of pixel processing algorithms of interest is based on using operators from mathematical morphology [37] as a means of segmenting vessels. A good example is the work of Zana and Klein [125, 126]. This algorithm is based on the idea that the vessels in a fundus image exhibit three identifying traits. First, the shape of the cross section is approximately Gaussian. Second, vessels are piece-wise linear and connected in a tree-like pattern. Third, they have a certain width and cannot be too close together. From these traits, morphological operators and algorithms are designed for classifying the vessel pixels. The first such possible operation would be to use the linearity trait. From this trait, a set of linear structuring elements can be used with opening and top-hat operations, which are discussed below.

Two morphological operations, termed “opening” and “closing,” respectively, are based on combinations of erosion and dilation operations. The opening operation is defined as an erosion operation followed by a dilation operation. The closing operation is defined as a dilation operation followed by an erosion operation. If the vessels are known to be darker than the background, a simple gray-scale dilation operation can be realized by a min-filter. Otherwise, a max-filter is used to dilate the bright vessels. For each pixel, the

max/min-filter output is the maximum/minimum value of all pixels within the neighborhood of that pixel. For dark vessels, erosion can be implemented by max-filtering.

Another morphological operation used is the “top-hat” operation. This operation is defined as the difference between the original image and the image resulting from an opening operation. This is useful in identifying the parts of the image that disappear as a result of the opening operation.

Recall the three vessel traits from above: the shape of the cross section is approximately Gaussian; they are piece-wise linear and connected in a tree-like way; and they have a certain width and cannot be too close together. Using the linear and width traits, it is possible to brighten/identify the blood vessels by summing the results of a series of top-hat operations that used a linear structuring element. Each top-hat operation would brighten a portion of the vessel structure that corresponded to the orientation of the structuring element. This structuring element would need to be long enough and oriented over a range of different directions to ensure that all vessels in all orientations would be eliminated by the opening operation in the top-hat operation and ultimately brightened (i.e., enhance the contrast between the vessels and the background) by the top hats. However, this process results in noise also being brightened. Another trait, the connectivity of the vessels, needs to be utilized before performing the top-hat operations.

To remove the noise in the image, a geodesic reconstruction [37] is first performed on the image. Performing this operation using twelve $15 \text{ pixel} \times 1 \text{ pixel}$ structuring elements, oriented at every 15° , results in an image $I'(x, y)$ where all elements in the original image $I(x, y)$ smaller than 15 pixels, such as noise and abnormalities, are eliminated. This operation is also known as a linear opening by reconstruction of size 15. Thus, the sum of the top hats can now be applied to brighten the vessels. At this point, a simple threshold can be applied to separate the vessels from the background but there still exist some background linear features or some bright or dark thin irregular zones that may falsely be identified as vessels. To eliminate these, the Gaussian-like curvature trait of the vessels can be exploited. This is done by computing the Laplacian of the Gaussian (LoG) at all points in $I'(x, y)$ and using its sign as an approximation of the sign of the curvature [101]. By using a Gaussian with a width of 7 pixels and $\sigma = 7/4$, the edges in $I'(x, y)$ can be detected based upon the curvature. If the sign of the curvature is positive, the pixels will be white and negative values will be portrayed in black. Once the LoG image $I''(x, y)$ is calculated, it once again becomes necessary to remove the noise in the background caused by “curves” detected in the previous process. This is done by applying three additional morphological operations in sequence to I'' . First, a linear morphological opening by reconstruction of size 15 is performed, followed by a linear closing by reconstruction of size 15 and a linear opening of size 29. The last step is to threshold the resulting image to determine the vessels.

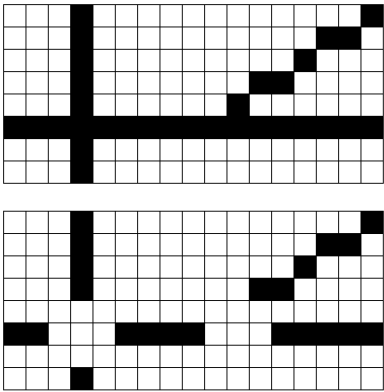


FIGURE 6.14
The breaking of the skeletonized vessels into segments.

Another pixel processing-based method, which can be thought of as a generalization of another morphological algorithm known as the watershed algorithm, is described by Hoover et al. [12]. The first step in this algorithm is to generate a set of seed points (or “queue points”) using the matched filter response (MFR) of Chaudhuri et al. [25] to enhance vessels over the background. They used the same matched filter given in Equation 6.7. This MFR is then thresholded where all pixels above the threshold are retained as candidate pixels. These pixels are then thinned to a 1 pixel wide linear structure that represents the vessels. The reader is referred to [3] for a more thorough discussion of thinning. From these thinned pixels, line segments are formed by erasing any point that has more than two adjacent candidate pixels. This, in essence, is deleting any point where a vessel splits into two, or the point where two vessels intersect while keeping any point that is an interior part of a segment as exemplified in Figure 6.14. When this is complete, all segments with fewer than 10 pixels are discarded and the remaining segment endpoints are placed into a queue as “seed” points. These seed points are then used as a starting point in a procedure called “threshold probing.” This is an iterative process where the threshold for an area is initially set to the intensity value in the MFR image of the seed pixel for the area. At each iteration, a region is grown from the seed using a conditional paint-fill technique. This paint-fill spreads to all connecting pixels whose values are above the current threshold and should be considered as “candidate” vessel pixels. Once all pixels for a particular threshold are identified, the region is then tested to see if any of the stopping criteria is met. If a criterion is not met, the threshold is decreased by one and the paint-fill technique is once again applied. The stopping criteria and its effect/rationale are as follows.

- *The number of pixels in the region exceeds a maximum number T_{max} .* This ensures that the region stops and that other regions are grown (and allows multiple threshold to be used for different regions of the image).
- *If the threshold reaches zero.*
- *If a region touches two or more previously detected regions.* This helps connect discontinuities on a vessel into a single “identified” region.
- *If a region is at least 30 pixels in size and is “fringing,” as defined by the ratio of the number of pixels bordering another region to the total number of points in region.* This ratio must exceed T_{fringe} . This prevents the current region from searching along borders of previously detected regions.
- *If a region is at least 30 pixels in size and seems to be branching too much.* This is determined using the ratio of the total number of pixels in region to the number of branches in the region. This ratio must be less than a threshold T_{tree} where the number of branches in a region is found by calculating the skeleton of the region. This prevents over-branching down false paths, particularly in poor contrast or noisy areas of an image.

Once the iterative probing is complete, the candidate region is labeled as a vessel if it connects two previous regions or has more than T_{min} pixels and less than T_{max} pixels. If the region is labeled a vessel, its endpoints are, in turn, added to the queue of seed points. When one of these “new” seeds is probed, an artificial boundary is placed perpendicularly at the end of the previous vessel region. Its purpose is to prevent the new probing from “fringing” on the previously detected region which forces the new region to grow away from the previous region.

6.5.1.2 Methods Based on Edge Detection

An algorithm described by Wang and Lee in [119] uses two basic approaches prevalent in edge detection. First, it performs an edge enhancement/thresholding technique to find vessel edges and then applies an edge fitting technique. For the first part of the algorithm, Sobel operators are used to enhance the edges of vessels. These edges are then thinned [3] so that the edge pixel with the lowest intensity value along each cross section of the vessel segments remains. These remaining edge pixels are used as “seeds” in the second part of the algorithm. As discussed in Section 6.4.7, it has been shown that one model of a vessel has a Gaussian-shaped cross section. Such edge models of the vessels can be obtained by local windowing and local thresholding, from which the skeletons of the vessels within the local gray-scale images are extracted [30].

The second part of the algorithm takes the local thresholded image and runs it through a set of twelve 15×15 matched binary filters (i.e., a filter consisting of only zeros and ones) with a single row of ones for each of the 12 cardinal directions (each 15° apart). These filters will yield a maximum response when aligned over the vessel boundaries found with the Sobel operators. After convolving these templates with every pixel, the mean and standard deviation can be calculated. Then for each seed pixel, the response of the 12 convolution values can be measured and compared against an experimentally determined threshold. Those responses that are higher than the threshold are considered to be vessel pixels. Those that are lower are considered background. The final step is to use a single linkage region growing technique with locally adaptive thresholds. Overall, this technique emphasizes speed by use of binary valued filters.

Li and Chutatape [67] implement a vessel detector using Kirsch operators in much the same fashion as [119] used Sobel operators and [25] used matched filters [67]. A Kirsch operator is a kernel that represents an "ideal" edge in each of the eight cardinal directions (0 to 360° , in 45° intervals) [101]. It works by applying each of the eight Kirsch templates at each pixel $I(x, y)$ in image I and building I' , where each pixel $I'(x, y)$ is the maximum response of the eight Kirsch templates at $I(x, y)$. A threshold is then experimentally derived and each pixel in I' greater than the threshold is considered a vessel.

Feature-based registration is one of the applications where segmentation is commonly used as a primitive. Pinz et al. have described a method for extracting vessels using matching edgels [85]. The purpose of their vessel segmentation algorithm is to extract "enough vessels" to be able to perform registration [9, 22, 23]. It works by first locating edgels (edge pixels) based on the gradient magnitude of a pixel and a locally computed threshold. The next step of the process is to find corresponding "partner" edgels. This is done for each edgel by first identifying a best matching "partner edgel." Potential partners are selected based on two constraints. First, the distance between the two must be between 3 and 17 pixels (inclusive) and, second, the angle deviation between the gradient direction of the edgels must be $180 \pm 30^\circ$. For each edgel in the set of potential partners, a cross-section line is used to connect them to the original edgel. The angular differences from the cross-section line to the gradients of the original edgel and the potential partner are computed and summed, and the cross section with the minimum sum determines the appropriate pairing. A common phenomenon known as vessel reflex (discussed in Section 6.4.8) necessitates another step in the process. This vessel reflex sometimes results in a single vessel cross section actually consisting of three pairs of connected edgels (cross sections). These edgels are combined into a single cross section if the orientations of the three are similar and the resulting cross section is below the maximum vessel diameter and the lengths of the two outer crossings are similar.

Once the vessel cross sections have been identified, the next step is to connect them and form vessel centerlines. This is done by connecting the midpoints of adjacent vessel cross sections as long as the following conditions are met:

- The cross-sectional lengths are similar.
- The distance between the cross section is below a certain value.
- The lines connecting consecutive cross sections should be relatively straight.
- The combined cross sections should be almost parallel.
- A cross section can only belong to one midpoint connecting line.

Once connected, short “vessels” are eliminated if the length of the vessel is smaller than the length of the cross section (i.e., the vessel width is larger than the length). The final step is to connect vessels located at crossings or intersections. These are the sites where only vessels are allowed to cross after checking both directions. This is done using the following constraints:

- The vessels to be connected must be longer than the distance to be spanned.
- The vessels to be connected must have similar cross-sectional lengths (vessel widths).
- The resulting connection must be as linear as possible.

Overall, pixel processing approaches are a useful tool, especially if computational speed is not an overriding concern. They can also be useful as preprocessors for the exploratory approaches discussed next.

6.5.2 Exploratory Algorithms

Approaches described in this section are referred to variously as vessel tracking, vectorial tracking, or tracing [21, 26, 109]. These methods work by exploiting local image properties to trace the vasculature starting from an initial point either specified manually or detected automatically. They only process pixels close to the vasculature, avoiding the processing of every image pixel, and so are appropriately called “exploratory algorithms.” They have several properties that make them attractive for real-time, live, and high-resolution processing because they can provide useful partial results [96] and are computationally efficient. (As an aside, numerous papers have been published on vectorization of binarized images within the document image processing literature; for example, [57]). These algorithms can be grouped in two categories referred to as semi-automated tracing where user input is needed and fully automated tracing where no user interaction is necessary.

6.5.2.1 Semi-automated Tracing Algorithms

One common type of semi-automated tracing algorithm is where the initial and end points of the vessel (sometimes also the direction and width) are entered manually. These are extensively used in quantitative coronary angiography analysis (QCA) [60, 102, 103, 109, 117, 127]. These algorithms are accurate but are unsuitable for real-time retinal image processing because they require manual input and suffer from high computational time, which is not a compelling constraint in QCA. Another QCA approach requiring initialization of points is the use of “snakes” [60, 104].

The first semi-automatic retinal algorithm is one used for identification and diameter measurement described by Gao et al. [45]. In it, two points c_0 and c_e on a vessel are manually supplied by the user. A line is drawn connecting these points with a direction d_0 . The region or neighborhood used for estimating the parameters of their model is limited to three times the estimated vessel profile width. The vessel is modeled as a sum of two Gaussian equations as shown in Equation 6.6, where its parameters are estimated by the nonlinear Levenberg-Marquardt method [87]. The seven unknowns ($a, b, c, d, \epsilon, \sigma_1, \sigma_2$) are varied from a predetermined set of values and the method works iteratively to minimize a χ^2 cost function to determine the parameters that best fit the profile. Once these best-fit parameters are found, the center point of the vessel can be determined from the value of a . Using this new center point, a second point is found in the same fashion using the perpendicular profile obtained at a point that is 2 pixels away in the previous direction (in this case d_0). This continues until six points are found. The centroid of these six points, c_1 , is the center of the first vessel segment whose direction is defined by the vector from the previous center (c_0) to the new center (c_1). This procedure of finding six points to determine the next segment continues until c_e is reached. Another algorithm where start and endpoints are entered manually for tracing retinal vessels in ocular fundus images is given by Pedersen et al. [83].

Another class of semi-automated tracing algorithms is where a manually entered initial point and direction is specified by the user. This tracing algorithm recursively tracks the entire arterial tree [68] using a breadth-first search. This would not be guaranteed to generate complete results for retinal images because the vessels are not necessarily connected, especially in partial views of the retina.

6.5.2.2 Fully Automated Tracing Algorithms

This set of algorithms automatically extracts the vascular tree without user initialization or intervention. They work well for coronary angiograms and have been applied to three-dimensional reconstruction [46, 79, 110]. Most of them [79] utilize the centerline gray-level intensities.

In retinal angiograms, although the blood vessels are darker than the local background, areas like the fovea are also dark relative to the average background. This consideration has motivated the development of algorithms

that rely on more localized cues, such as contrast and edge gradients, similar to some of the methods used in QCA [21, 102, 103]. This also enables the algorithms to be more robust to lighting-related artifacts such as glare, dropouts, and overexposed frames that can easily occur in retinal images, especially in images containing pathologies that are generally of most interest.

A Kalman filter-based method that uses Gaussian matched filters is described in [26]. This method utilizes an amplitude-modified, second-order Gaussian model for the blood vessels. The first step of this algorithm is to extract an intensity profile from the circumference of the circular region placed around the optic disk (found using a separate algorithm). This profile is then convolved with the amplitude-modified, second-order Gaussian filter described by Equation 6.5. The points at which the response is a relative maximum are candidate points at which vessels are entering/leaving the optic disk. From this set, the six highest maxima are selected as vessel starting points. These points are placed into a queue and are selected in turn for vessel tracking.

Each vessel segment is defined by three parameters: the width w , the centerline point c , and the direction d . For each step in the tracking process, a Kalman filter [51] is used to give the optimal estimation of the next centerline point (based on the previous location, direction, and supplied tolerances/covariance). At this estimated location, the Gaussian filter, as described in Equation 6.5, is applied to find the true center. In this process, a value of $w/2$ is used for σ , where the w is the value of the width of the last vessel segment. Once the correct location is found, the new segment's location, direction, and width are updated. The zero crossings of the response are then used to determine width (i.e., width = distance between zero crossings/1.2). At each step, if a branch is detected, its location is added to the start point queue. "This iterative process ends when an endpoint is detected by Gaussian filter and the tracking result is recorded" [51].

While this algorithm is successful at tracing the vessels, its success strongly depends on the initial estimation of the optic disk, which in itself is a hard problem [66]. Additionally, by using the optic disk as the starting location, this algorithm is restricted to the set of retinal images in which the optic disk is present. Another issue not addressed by this algorithm is the fact that all the vessels in the image are not necessarily connected to the optic disk. Another automated tracing algorithm that uses steerable filters is given in [62]. This algorithm also uses the optic disk to initiate the traces and suffers from the same limitations.

Another exploratory algorithm based on vessel models that are based on local gray-scale intensity differences is described by Collorec and Coatrieux [31]. As can be expected, this issue leads to some "hard problems" as noted by Coatrieux et al. [29]. Specifically, they note the difficulty in handling branching and crossover points because they do not conform well to the vessel models, the difficulty of making the algorithms locally adaptive, and the problem of wandering and looping of traces.

6.6 RPI-Trace

Can et al. [21, 96] have addressed the issues noted by Coatrieux et al. using a stricter model for the vessels in an algorithm. Their algorithm and implementation gave birth to what is referred to as RPI-Trace in this chapter. Can et al. overcome two “hard problems” described by Coatrieux et al., namely, (1) robust and accurate handling of branching and crossover points; and (2) improved handling of discontinuous regions by relying on local contrast and edge information (as opposed to gray values), instead of a global intensity threshold. It also overcomes the “wandering and looping” artifact using an improved stopping criterion. Some tracing accuracy improvement is also gained by more accurate angular discretization, more filtering across and along the vessels, and more careful handling of the discrete image space. Computationally, this algorithm is comparably attractive. The very strict rules used by this algorithm for validation and verification of initial seed points allow it to reject many incorrect seed points, thus making up for the higher complexity in the core tracing computations. More recent work has resulted in sub-pixel localization of vessel boundaries, and much improved handling of branching and crossover points.

Can’s method relies on a recursive tracing of the vasculature based on a localized model. This approach has been shown to be much faster (e.g., video frame rates are readily achieved), more adaptive, and more practical for implementation on conventional and parallel MIMD computers [4]. It also requires the fewest and provides useful partial results [96], which is discussed in Section 6.7. This fully automated exploratory tracing algorithm proceeds in three stages.

Step 1 (seed point initialization): The algorithm analyzes the image along a coarse grid (see [Figure 6.9a](#)) to gather gray-scale statistics (contrast and brightness levels) and to detect seed locations on blood vessels using the gray-scale minima between opposite-signed, one-dimensional edges described in Section 6.4.5. False seed points are filtered out by testing for the existence of a pair of sufficiently strong parallel edges with opposite gradient values. For this, a set of directional kernels is applied to the seed’s neighboring points, radially in 16 directions, in which the kernel used is orthogonal to the radial line, and the two strongest responses are found. This search strategy is illustrated in [Figure 6.16](#). The initial point is filtered out if the two strongest responses do not both exceed a sensitivity threshold T_e given in Equation 6.3, or if the directions of the two strongest edges are not sufficiently similar (within $\pm 22.5^\circ$). On average, about 40% of the initial points are filtered out by this procedure.

Step 2 (recursive tracing): The second stage, illustrated in [Figure 6.15](#), is a sequence of recursive tracing steps that are initiated at each of the filtered seed points, and proceed along vessel centerlines using an update equation

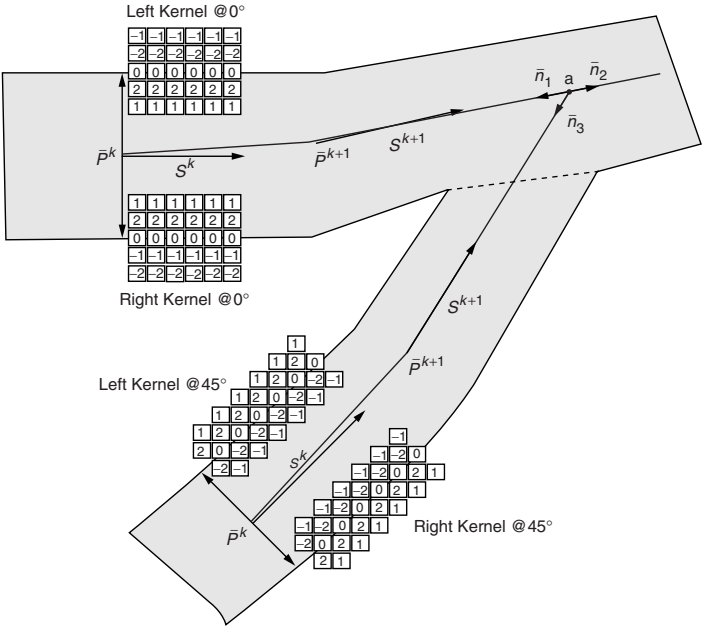


FIGURE 6.15
Illustration of the core tracing step in the exploratory algorithm of Can et al. [21]. The left and right kernels are shown for 45° and 0° on the two left vessel branches. Starting from a point p^k , the direction of maximum response of these kernels defines the direction of the tracing step. Crossing and branching points of the vasculature are detected where traces intersect.

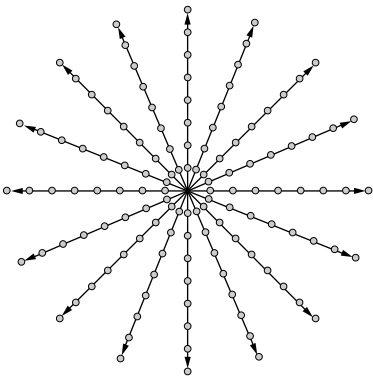


FIGURE 6.16
The radial search strategy used to verify initial points as vessel points at which to initiate tracing in Step 1. Each dot represents a location at which a template perpendicular to the radius is applied to search for an edge.

of the following form:

$$p^{k+1} = p^k + \alpha \left[\cos\left(\frac{2\pi s^k}{N}\right), \sin\left(\frac{2\pi s^k}{N}\right) \right]^T + \beta^k \quad (6.8)$$

where k is the iteration count, p^k and p^{k+1} denote the current and new locations of the trace, α is a step size, $s^k \in \{0, 1, 2 \dots N-1\}$ is an integer index specifying one of N discretized angular directions (usually $N = 16$), and β^k is a lateral displacement vector that centers the new point p^{k+1} on the vessel. In Figure 6.15, this is illustrated for a pair of intersecting vessels. The left and right directional kernels at 0° and 45° are also illustrated for the lower branch in Figure 6.15. The angle at which the correlation kernels produce the highest response is estimated. These maximum responses are computed by performing a local search along a line perpendicular to the current trace within a neighborhood of size $M/2$, where M is defined as the maximum expected blood vessel width. The next trace point p^{k+1} is determined by taking a step of size α , from the current trace point in the direction s_{k+1} , and applying a correction β that is calculated based on how far p_{k+1} was from the center of the points at which the left and right template responses were found. This new point is kept only if it does not intersect any previously detected vessels, is not outside the image frame, and if the sum of maximum template left and right responses is greater than the sensitivity threshold, which is computed as follows:

$$T = 6L(1 + \lambda|L_{av} - B_{av}|) \quad (6.9)$$

where L is the length of the template, and L_{av} and B_{av} are the estimates for the average vessel intensity and average background intensity, respectively. Hence, their difference is an estimate of average contrast. λ is a scaling factor between 0 and 1 that can be thought of as a percentage of overall contrast, and the 6 is two times the response of the template for a ridge of one unit high per unit length (i.e., response per contrast increase). Low values for λ make tracing more sensitive to noise whereas high λ values cause tracing to terminate prematurely. Average intensity values for the background and vessel are estimated using the intensity values along the grid lines used for seed point detection.

Step 3 (landmark extraction): The tracing that starts from a seed point continues until the end of the vessel is reached, or until the centerline of the tracing intersects a previously detected vessel centerline. Landmarks are placed at intersections of traces and at locations where three or more centerline traces meet, as shown in Figure 6.30f. In the former case, the location is the actual intersection; in the latter, the location is the centroid of the trace endpoints. These landmarks are characterized by this location and by the orientations of the traces meeting to form the landmark. Example trace results are shown in Figure 6.17, and a composite result containing traces and landmarks is shown in Figure 6.2.

Modifications and improvements to RPI-Trace have been implemented in [93] and by the authors of this chapter. These modifications include:

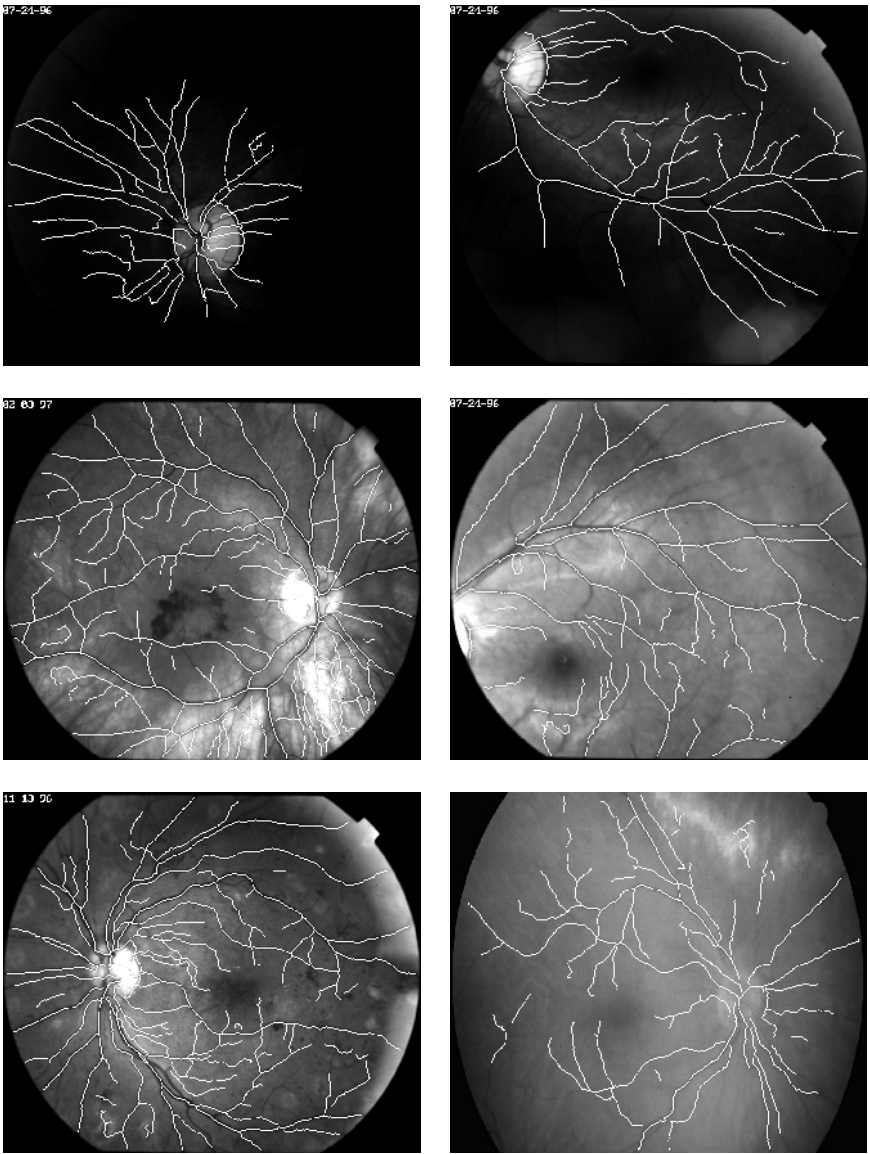


FIGURE 6.17
Results of exploratory tracing [21] on a diverse set of images with varying lighting conditions and presence of pathologies.

- The centerline, boundary, and seed point locations are computed with sub-pixel accuracy. The boundary locations are calculated to sub-pixel accuracy by fitting a parabolic model to the template response around its maximum, and then finding the location of the peak from the center of this parabola. For certain applications, sub-pixel computation of image centerlines and landmarks gives improved precision for higher-level computer vision tasks such as registration and spatial referencing [94, 95].
- The seed points along the grid lines are determined more accurately. This is achieved using one-dimensional edge detection constraints in addition to intensity constraints and local thresholds, as described in Section 6.4.5. Results of this method are shown in [Figure 6.10](#).
- The local threshold value for T_e , as described in Section 6.4.5, is used in finding each trace point rather than a global threshold.
- Instead of searching for the boundary locations starting at the current trace point and stepping out, the width of the previous locations is used to limit the search space for the boundary location at the current iteration. Based on the current iteration's estimated location of a vessel boundary, the search space is bounded within a fixed distance Δ . Hence, abrupt changes in boundary location between two iterations is prevented and the computational time is reduced.
- Instead of applying a set of templates with fixed length, multiple templates of different lengths are applied, with the resulting responses being normalized based on template length. Detection and localization can be improved simultaneously by longer directional templates [24].
- The stopping criteria are enhanced by only terminating the tracing when several consecutive weak responses are detected, rather than a single weak response. This modification gives additional robustness to the tracing.
- The use of a bin data structure to calculate intersections is added. Two spatial data structures, each representing the area of the original image are created. One is used to track center points and the other is used to track boundary points. In each of these structures, the area of the original image is broken into smaller pieces or "bins." Each bin represents an equivalent area on the original image, typically a 4×4 or 8×8 pixel area. When a center point or a boundary point is found, it is "placed" into the bin that corresponds with its location in the original image. When it is desired to see if a new center point intersects with an existing trace, the appropriate bins are queried. These bins help to reduce the time needed to determine if there is an intersection and can also be used in other algorithms such as registration and spatial referencing.

6.7 Algorithm Design Considerations for Real-Time Tracing

The design of algorithms for real-time vessel tracing are driven by the end objective. Spatial referencing [95] is an excellent application example. Here, the end goal is not the vessel traces by themselves, but rather the results of image registration. The role of the vessel traces is to provide the spatial landmarks needed for the registration. This is an instance of a “hard” real-time system [65] in which the computations must be completed prior to a deadline, or else the system is considered to have failed. A failure represents a loss of tracking, requiring the surgical laser to be switched off [122]. System performance degrades with an excessive number of failures. The computational deadlines are dictated by the frame rate of the imaging camera (usually about 30/sec). Within this deadline, two computations must be completed: landmark extraction and landmark-based image matching. For a given computer system, the computational budget is fixed within this deadline. It is desired to maximize the probability of a successful image match within this budget.

The computational budgeting issues are illustrated in Figure 6.18. The left and right columns show the partial results of two different tracing procedures, captured at 8%, 28%, and 53% of the total computational effort. The only difference between the two procedures is the order in which the vessel segments are traced; they use the same tracing algorithm, the same total amount of computation, and produce the same final result. The partial results in the right column are far more valuable because they have more numerous and more prominent landmarks. Given a high-quality partial result, image matching can be attempted from partial traces. If this is successful (i.e., sufficient confidence exists in the result), then the overall system can be much faster. Even if this is unsuccessful, it can form the basis for a better subsequent matching attempt. The frame cycle time can be subdivided into a series of milestones and the performance of the matching algorithms monitored at each milestone. Failure to reach set milestones can provide an early indication of conditions such as poor image quality (images that are dim, saturated, out of focus, affected by glare, etc.). The idea of utilizing partial results relates to the concept of imprecise computations as proposed by Lin et al. [75] within the real-time scheduling literature. The methods for scheduling the tracing computations in a manner that produces such an early yield of high-quality partial results are described next.

The spatial prioritization scheduling algorithm of Shen et al. [96] aims to maximize the number and quality of the bifurcation and crossover points in the image within a given number of computations. The quality of an individual landmark can be quantified based on the prominence of the intersecting vessels and their contrast, as follows. Suppose a single landmark is formed by the intersection of m vessel segments. Let t_p be the estimated thickness

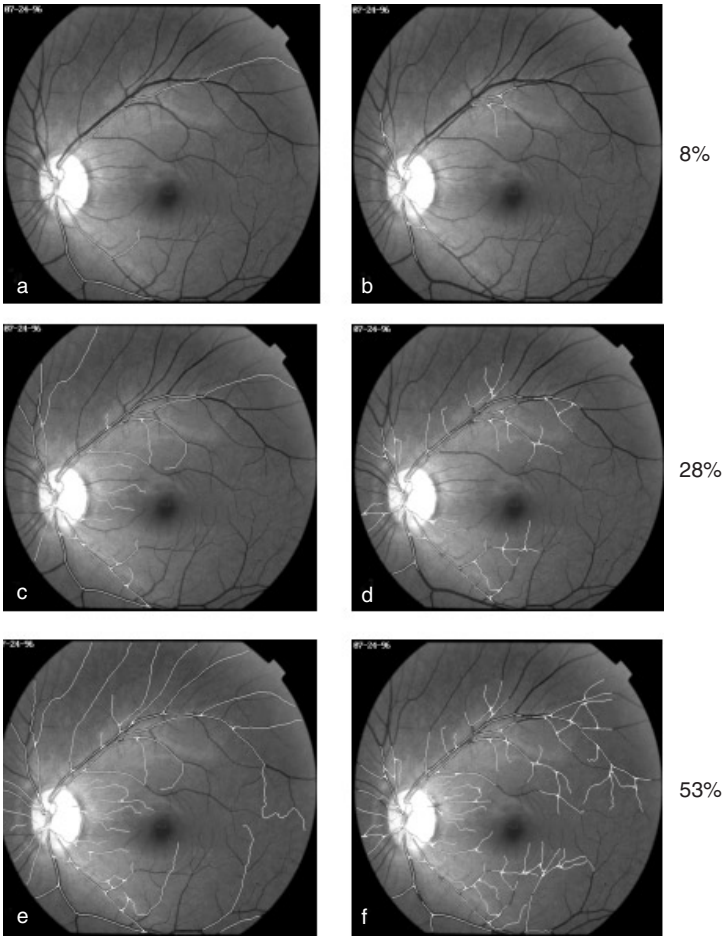


FIGURE 6.18

The impact of scheduling on landmark detection performance for the same level of computational effort (W/W_{total}): (a) a partial result from tracing an image using a poorly scheduled algorithm shows the work wasted on long and unproductive tracing; (b) a partial result from a better scheduled algorithm for the same computational work. The latter is much more productive, having prioritized its efforts in a manner that yields numerous prominent landmarks early. (c-f) Partial results for the two algorithms at 28% and 53% effort.

of segment p , and s_p a measure of its edge strength. Then, the quality of the landmark can be measured as:

$$q = \sum_{p=1}^m s_p \cdot t_p \tag{6.10}$$

The edge strength s_p is estimated using the directional correlation kernels [21] and summarized in the previous section. The quality of a partial result

is measured simply by summing the qualities of the individual landmarks constituting it.

The computational effort can be quantified as well. One straightforward method to quantify computational work is to count the processor cycles. However, such a measure is necessarily dependent upon a specific combination of processor, compiler, and other implementation details. An extensive profiling analysis revealed that the vast majority of the processor cycles were expended on computing the response of local correlation kernels. This suggests a simple system-independent measure of computational work: simply count the number of correlation kernel computations. We use the variable w to denote this count. The tracing of the vasculature is carried out through a sequence of computations, with the work (as measured by counting correlation kernels) denoted w_1, w_2, \dots, w_N , that produce a sequence of non-negative incremental results q_1, q_2, \dots, q_N . The quality of the partial result at the completion of w_k is given by:

$$Q_k = \sum_{n=1}^k q_n \quad (6.11)$$

where q_n is the quality measure defined in Equation 6.10 for the n^{th} landmark. The cumulative computational effort used to produce the above partial result is given by:

$$W_k = \sum_{n=1}^k w_n \quad (6.12)$$

At this point some observations can be made. The sequence of partial results is always of monotonically improving quality; that is, $Q_{k+1} \geq Q_k$. In the terminology of imprecise computation, the tracing algorithm is an “anytime algorithm” [75]. The amount of vasculature in the (finite-size) image is finite. Thus, a definite endpoint exists for the tracing, and the last partial result also represents the complete result. For a given image, and algorithm settings such as the grid size and sensitivity, the quality of the complete result, denoted Q_{total} , and the total computational work W_{total} are both finite and fixed. Finally, the order of tracing of vascular segments has minimal effect on the final result. The above observations suggest that, in principle, one could divide the vasculature arbitrarily into vessel segments and decide the order in which the vessel segments are traced. A scheduling algorithm can decide the starting, stopping, and restarting of the tracing, and choice of seed points. An optimal scheduling algorithm will make decisions that maximize the rate at which the quality of partial results improves.

The monotonically improving quality of a partial result can be described by a growth function $Q(W)$. This function does not have a closed-form expression, and is represented by an empirically derived Q vs. W curve. The quality and total work (Q_{total} and W_{total}) are different for different images, so for comparison and averaging across images, it is advantageous to normalize

Q and W by Q_{total} and W_{total} , respectively. With this in mind, the question of interest is: given a partial computation $W_k/W_{total} \leq 1$, how can the tracing computations be scheduled so that the normalized quality of the corresponding partial result, Q_k/Q_{total} , is maximized for every k ? This can also be viewed as maximizing the area under the normalized Q vs. W curve. The fully optimal scheduling algorithm must perform this maximization globally for every partial result — a daunting task. To maximize the optimality criterion just described, the scheduling algorithm must somehow perform just enough tracing to detect each of the landmarks, thereby minimizing the amount of work done. Simultaneously, it should trace around the highest-quality landmarks first, thereby maximizing the quality measure. Unfortunately, achieving this objective requires prior knowledge of the very traces that are sought — an inherently impossible task. This leads to the inevitable conclusion that an optimal schedule can only be computed in hindsight or, equivalently, with perfect foresight. In other words, the optimal schedule is hypothetical. Even with perfect foresight, computation of the optimal schedule is a difficult global combinatorial optimization; for each value of $W_k \leq W_{total}$, one must select a set of landmarks so that the sum of their computational work does not exceed W_k , and the sum of their qualities is the maximum.

Shen et al. [96] described an approximation to the optimal solution known as the preemptive spatial prioritization algorithm. A detailed step-by-step dissection of the optimal schedule for several images reveals that it primarily derives its high performance from advance knowledge of the spatial locations and quality values of the landmarks. It concentrates just-sufficient levels of tracing effort around the most promising landmarks. This observation suggests predicting landmark locations and qualities, and using these to schedule the tracing.

Landmark locations and qualities can be predicted from the initial gridanalysis step of exploratory tracing. The parallel edge-based model discussed in Section 6.4.6 can be used to weed out all seed points that do not correspond to a pair of nearly parallel edges, and produce local orientation, strength, and width estimates (Figure 6.19a). Shown in Figure 6.19b through g are cropped and enlarged views of selected boxes formed by the grid lines. In these images, the short lines crossing the grid lines mark the estimated orientations of the initial points. The black dots on these lines indicate the location of the initial points after the filtering. From these images, the presence/absence of landmarks inside the grid box can be guessed even before tracing by analyzing the distributions and orientations of the initial points around the grid box. For example, in Figure 6.19b through d, the orientations of the initial points clearly suggest the existence of at least one landmark inside the grid box. Conversely, in Figure 6.19e and Figure 6.19f, although there are initial points associated with that grid box, their parallel orientations suggest that there is probably no landmark in the grid box. However, not all grid boxes exhibit the simplicity of Figure 6.19b through Figure 6.19f. For instance, the

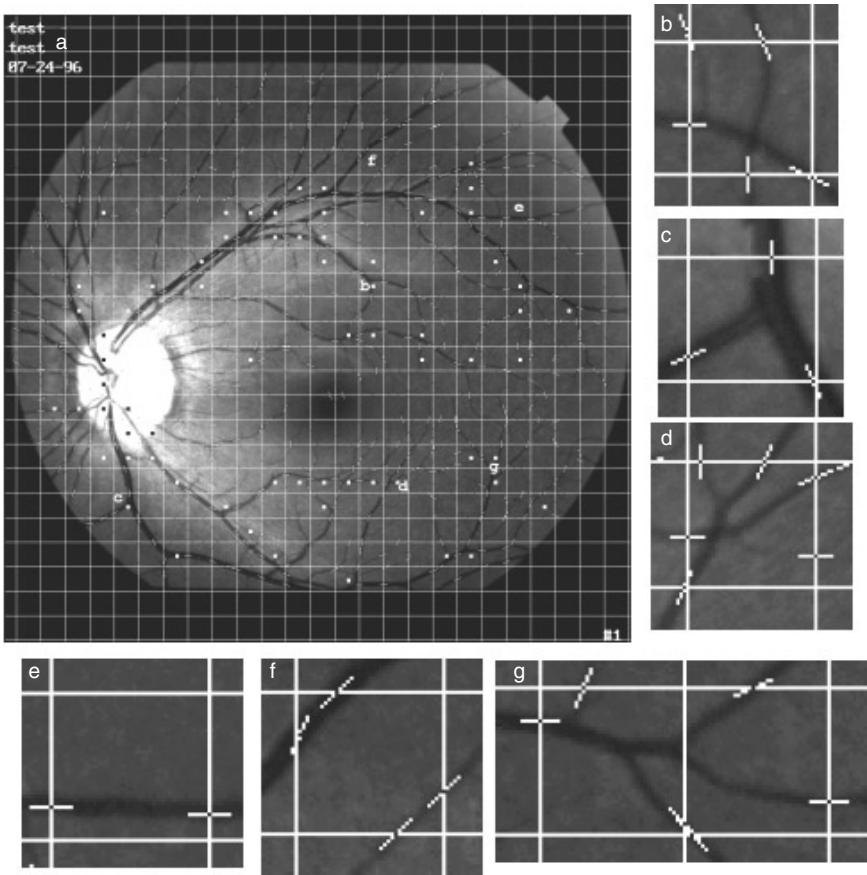


FIGURE 6.19

The basis for spatial prioritization. (a) The small line segments crossing the grid lines indicate the filtered seed points. Their orientations indicate the estimated local vascular orientations. Panels b through g highlight selected regions from panel a. Boxes with the highest 10% of $D(G)$ values are shown marked with a dot in the middle. Panels b through d illustrate cases when the filtered initial points provide strong clues about the presence of landmarks within the grid box. Panels e and f illustrate cases when the initial points suggest the absence of landmarks. Panel g illustrates a more complex case. Although there is no landmark inside this box, a landmark is right beside its left border. This example suggests that one should not only consider the possibility of a landmark inside a box, but also in the box's neighborhood. Notwithstanding such cases, this approach is very successful.

right-hand box in Figure 6.19g shows a more complex case. Although there is no landmark inside this box, a landmark is next to its left border. This suggests that one should not only consider the possibility of a landmark inside a box, but also its neighborhood. On the other hand, errors in seed point

detection or filtering can produce misleading conclusions about grid boxes. A computationally simple yet effective measure is needed to capture these intuitions. Primarily, it must rate the likelihood of a landmark occurring inside a grid box and estimate its quality. We choose a weighted angular diversity measure adapted from angular statistics [120]. The angular variance of a set of unit vectors $\{\vec{u}^{(1)}, \vec{u}^{(2)} \dots \vec{u}^{(K)}\}$ is defined as $1 - \frac{1}{K} \|\sum_{k=1}^K \vec{u}^{(k)}\|$, where $\|\cdot\|$ is the standard Euclidean norm. Intuitively, the length of the resultant of the unit vectors is as large as possible K when all the vectors are aligned with each other, and the least possible 0 when they are pointing in opposite directions, cancelling each other out. For the present work, a strength-weighted version of the angular variance is used, and is further weighted by the total number of vectors. Let the number of initial points on a grid box be K . Let the two-dimensional estimates of edge strength be $\hat{s}_{2D}^{(k)}$, $k = 1, 2, \dots K$, and let the unit vectors $\vec{u}^{(k)} = [\cos \theta^{(k)}, \sin \theta^{(k)}]^T$ indicate the local vessel orientation estimates. Then, the strength-weighted angular diversity measure for a grid box G is:

$$D(G) = \sum_{k=1}^K \hat{s}_{2D}^{(k)} - \left\| \sum_{k=1}^K \hat{s}_{2D}^{(k)} [\cos 2\theta^{(k)}, \sin 2\theta^{(k)}] \right\| \quad (6.13)$$

The angles must be doubled because the angular diversity between a pair of vectors is maximum when they differ by $\pi/2$, and minimum when they differ by π . In computing the above measure, an initial point is assigned to a grid box if it is located on or sufficiently close to any of the grid lines that form the grid box. Therefore, an initial point may belong to more than one grid box. The grid boxes are prioritized by their values of $D(G)$. Boxes with more numerous and stronger initial points and with a higher angular diversity will have greater values of $D(G)$. To illustrate the effectiveness of this approach, the top 10% of the grid boxes in the priority queue for the image shown in [Figure 6.19a](#) are indicated with dots in the center. Note that most of these marked boxes have at least one landmark within or nearby.

Within a grid box, the initial points are prioritized equally. The tracing is preempted after a number of steps that is proportional to the size of a grid box. Upon preemption, the stopping point is inserted back into the priority queue at the same level of priority as the grid box in which the tracing was stopped. An exception occurs when the tracing is preempted in a grid box whose priority is higher than the grid box from which the tracing was initiated. In this case, the tracing is continued until the next preemption. Therefore, this is a preemptive scheduling algorithm with both spatial- and edge strength-based prioritization. [Figure 6.20](#) shows the quality of the partial result Q as a function of computing effort W for a typical image. Overall, this algorithm produced $\approx 400\%$ improvement in the quality of the partial results at a defined milestone (33% of the total tracing), compared to random scheduling.

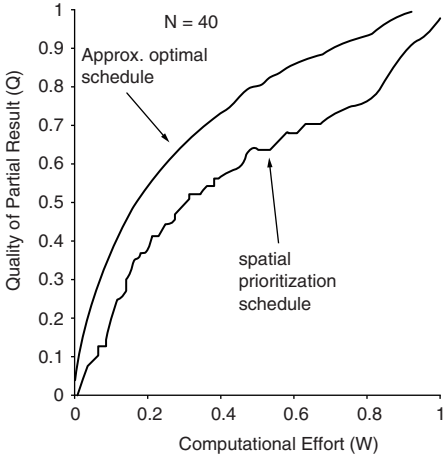


FIGURE 6.20

The growth of the quality of the partial tracing results as a function of the computational effort as measured by the number of kernel operations. The higher curve is an estimate of the optimal schedule, which is hypothetical and unachievable. By mimicking the observed behavior of the hypothetical scheduler, it is possible to approximate its behavior using the preemptive spatial prioritization scheduling algorithm.

6.8 Accurate Extraction of Vessel Bifurcations and Crossovers

Branching and crossover points in neuronal/vascular structures are of special interest from the standpoint of biology and medicine [9, 63]. One such application is the early diagnosis of hypertension by measuring changes in select vascular branching and crossover regions [82, 114]. Another example is the study of early development of the retinal vasculature, and its evolution under various pathologies and applied conditions [39, 77, 80, 97, 112].

Branching and crossover points are also important from a purely image analysis standpoint. The locations of these points, if known to be stable, are valuable as features (i.e., landmarks) for image registration and mosaicing [32, 33, 43, 124]. The pattern of angles of intersection can be used as landmark signatures [9, 10, 22, 23].

Crucial to the performance of image registration algorithms is the accuracy and repeatability with which vascular crossing and branching locations (landmarks) can be extracted, more so than their absolute location [9]. The landmarks, when placed in correspondence, constrain the image-to-image transformation that must be estimated to register the images. Of particular interest, landmark repeatability plays two crucial roles: to reduce the number of possible correspondences between two images and accurately initialize a

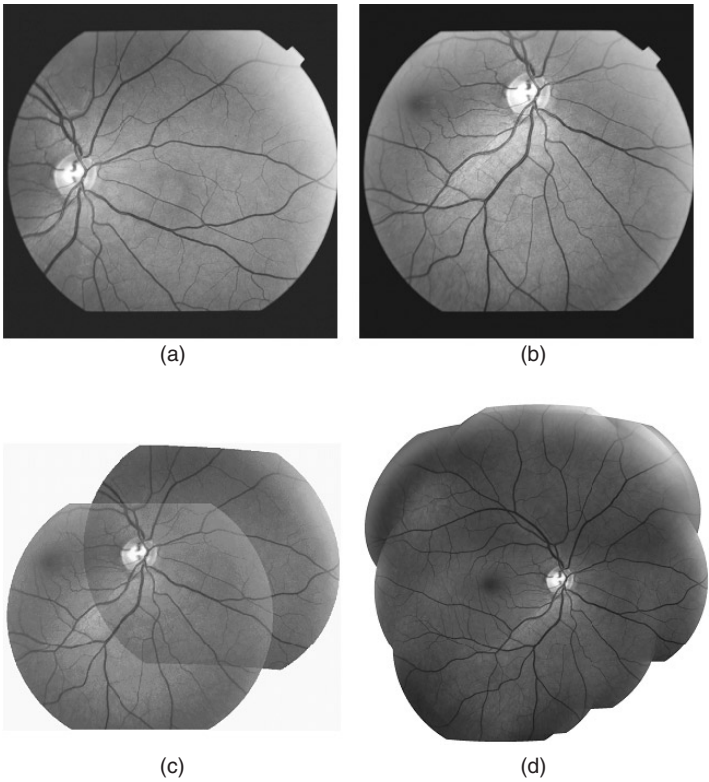


FIGURE 6.21 Retinal image registration and mosaicing. Panels a and b are fundus images taken at the same time with approximately 50% overlap (c) the results of registering panels a and b, displayed as a mosaic. Panel d is a complete mosaic of the same retina. Accurate registration to sub-pixel accuracy requires precise and repeatable estimation of image features (landmarks), and their signatures (intersection angles and thickness values).

local transformation for registration. The more repeatable the landmarks, the more likely the registration process is to succeed with the least computation.

The design of the landmark extraction technique in the exploratory tracing algorithm in [21] is conceptually simple and effective in terms of detection. However, it suffers from limitations relating to the accuracy and repeatability of estimating the intersection coordinates and angles.

These limitations arise primarily from the fact that the anti-parallel edge model on which the tracing algorithm is based, is no longer valid very close to branching and crossover points due to the rounded nature of the junctions (see Figure 6.22). Consequently, when the recursive tracing steps approach a junction, the estimation of the centerline of the vessels is less accurate. That is, trace points become inaccurate very close to intersections. The peak responses of the left and right templates often occur for many different orientations as they

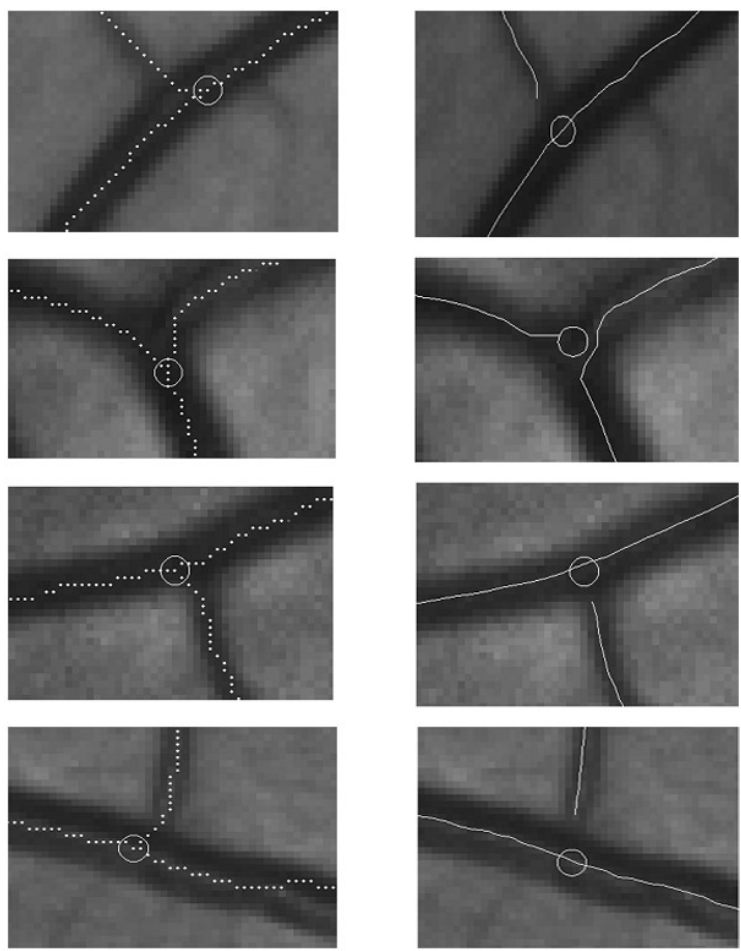


FIGURE 6.22
The issue of landmark location accuracy. The left column shows enlarged close-up views of three landmarks, overlaid with the results of tracing from our earlier algorithm. As the tracing steps approach the intersection, the anti-parallel edges model that holds well for the straight portions of the vasculature fails, leading to errors in estimating the intersection position and angle. The column on the right shows results produced by the enhanced algorithm (ERPR) presented in this section. This algorithm estimates the locations of the intersections, and the angular signatures more accurately using a model. The detected intersection is the center of the overlaid circle.

begin to overlap intensity values from two or more different blood vessels. This may result in uncertain and unreliable placement of centerline points. Finally, errors in estimating the point of intersection have a pronounced effect on the accuracy with which the intersection angles are estimated. The issues related to landmark accuracy are illustrated in Figure 6.22. The issues related to repeatability are illustrated in [Figure 6.23](#).

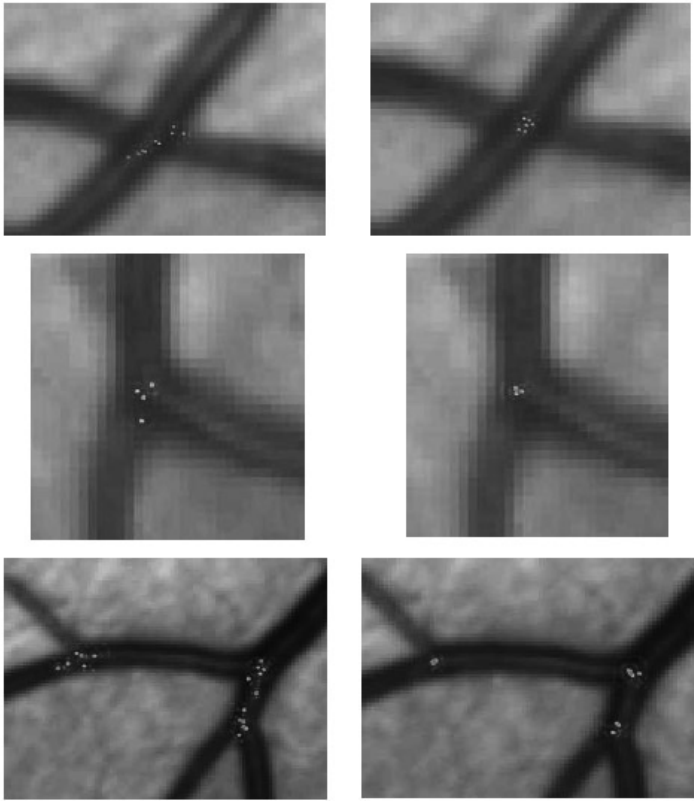


FIGURE 6.23
The issue of landmark repeatability. The dots on the vessels are the landmarks mapped from the fundus images. The left column shows three examples from the original method. The right column shows the corresponding results from the ERPR algorithm for the same image regions. Observe the substantial improvement in the repeatability with which the locations are estimated. This is important for image registration applications, especially with real-time implementations.

Our discussion here leaves out nonexploratory approaches for locating the branching and crossing points using interest operators [9, 59, 72, 76].

6.8.1 A Model for Vessel Bifurcations and Intersections

The failure of the anti-parallel edges model near bifurcations and intersections is the primary source of error. The solution described here is to build an explicit model of the structure of a landmark and an estimation technique that estimates the parameters of this model. The proposed landmark model, illustrated in [Figure 6.25](#), consists of three parts:

1. *A circular exclusion region.* This region models the region of intersection of the blood vessels. In this region, the anti-parallel model of the blood vessels is violated. Therefore, traces computed in this region are not used.
2. *The landmark location.* This is defined as the (x, y) point nearest the extrapolation of the centerlines of the vessels that meet to form the landmark.
3. *Orientation vectors.* The set of blood vessel orientations that meet to form the intersection. These orientations are defined relative to the landmark location.

The exclusion region radius is estimated once, but the other parameters — the landmark location and the blood vessel orientations — are estimated iteratively. The following subsections provide a more detailed description of these items, the estimation algorithms, and the relevant notation.

6.8.1.1 Overview of the Landmark Refinement Algorithm

The starting point for the estimation process is the endpoint of a trace when either it intersects the boundary of another blood vessel or it meets at least two other trace endpoints. From this endpoint, the algorithm gathers information about neighboring traces, estimates the initial landmark point, determines the exclusion radius, and estimates the initial blood vessel orientations. This initializes an iterative process that alternates steps of re-estimating traces and orientations of blood vessels outside the exclusion region, and then re-estimating the landmark point from the blood vessels. Together, these steps ensure that the traces outside the intersection are more accurate, that multiple trace points are used in estimation, and that ultimately the final landmark location is estimated more accurately.

6.8.1.2 Gathering Information About Neighboring Traces

The estimation technique for a single landmark starts from a trace endpoint and a set of neighboring traces. Figure 6.25 illustrates the terminology and notation used here. A *trace* is defined as a sequence of centerline points detected during recursive tracing starting from a single seed point. Let T be the set of all traces (across the entire image), $t \in T$ be a single trace, and $P(t)$ be the sequence of centerline points on trace t . The centerline points on trace t are $p_{t,i}$, and $p_{t,e}$ is the ending centerline point (“endpoint”). The set of neighboring traces, denoted by $N(p_{t,e})$, is the set of other traces having at least one centerline point close to endpoint $p_{t,e}$. Finally, let $w_{t,i} = w(p_{t,i})$ be the width of the blood vessel at each centerline point, computed easily during recursive tracing.

6.8.1.3 Initializing the Landmark Model Parameters

During the initialization process, the landmark location, the exclusion region radius, and the blood vessel orientations are estimated in turn. The exclusion region radius remains fixed throughout the computation.

Initializing the landmark location starts by gathering sets of neighboring traces' trace points. These sets are formed for each trace's end point by searching an area with a radius of 10 pixels for nearby traces. When a nearby trace is found, if its closest trace point is an endpoint, it is included in the set. If the trace's closest point is a mid-point, then the trace is split into two separate traces and the two new endpoints are added to the set. For each set:

- If the set has only two traces, then no landmark is placed here.
- If the set contains only endpoints (at least three), then the initial landmark location, denoted by q^0 , is the centroid of the endpoints.
- If the set was made from splitting a trace, the initial landmark location q^0 is set at the split point p . See Figure 6.24a for an example of this.

Next, the exclusion region radius r^* is estimated. Intuitively, the exclusion region should have a diameter at least as wide as the width of the thickest vessel in the intersection. Because the width of a vessel varies, and is less reliable as the vessel approaches the intersection, we define the width of a trace $w(t)$ as the median value of widths of all trace points on t . The exclusion radius r^* is then defined as the maximum of the trace widths for all traces from which a landmark is made. There is no need to refine r^* because it does not depend significantly on the landmark location or other landmark parameters.

The final step in initialization is estimating the blood vessel orientations near the intersection. This orientation is denoted $\phi(t)$ for each trace at the landmark. The initial value of this orientation $\phi^0(t)$ is found by fitting a line to points on trace t that are just outside the exclusion region. This is described in more detail below because it is exactly the same computation as used in the iterative procedure.

6.8.1.4 Iterative Estimation of Model Parameters

The trace centerlines, blood vessel orientations, and landmark location are estimated iteratively. The first step in each iteration $j \geq 1$ is to re-estimate the trace centerline points near landmark location q^{j-1} , but outside the exclusion region. This procedure is called *back-trace refinement*, and is illustrated in Figure 6.24. For each trace, a seed point on the boundary of the exclusion region is found, and then the recursive tracing procedure (described by Can's algorithm in Section 6.5.2.2) is run for a small number of steps (e.g., $n = 5$) away from the intersection. The seed point for each trace is simply:

$$q^{j-1} + [r^* \cos \phi^{j-1}(t), r^* \sin \phi^{j-1}(t)] \quad (6.14)$$

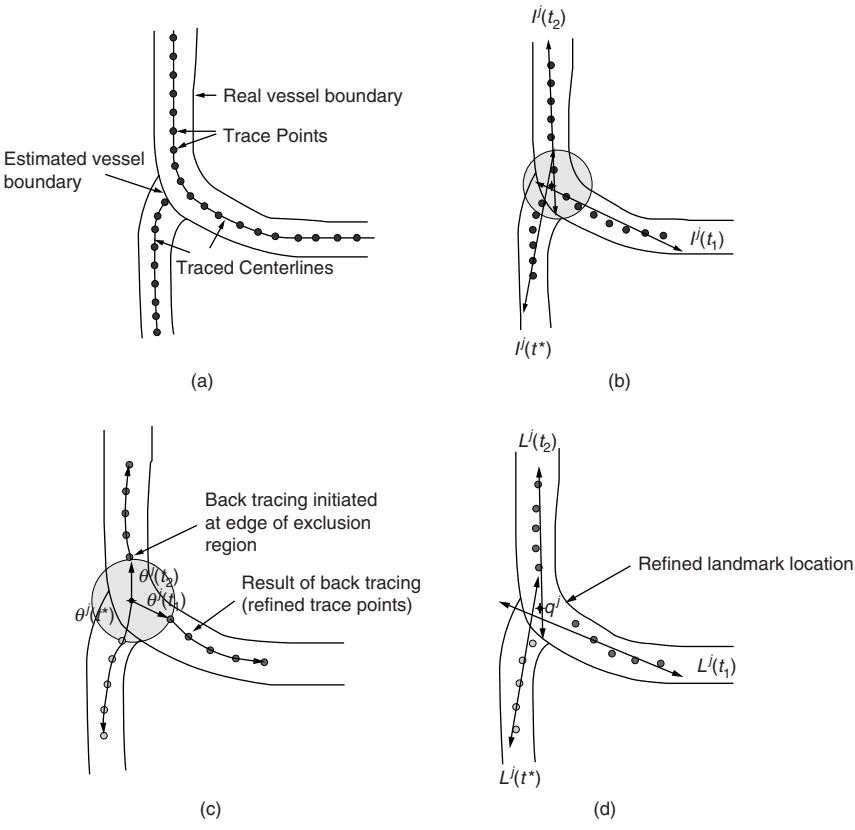


FIGURE 6.24 The steps in estimating model parameters: (a) the result of initial tracing; (b) showing the exclusion region (circle) and method for initializing the back-tracing, lines, denoted $l'(t_1)$, $l'(t_2)$, and $l'(t^*)$ are fit to the previously traced centerline points; (c) showing new traces from back-tracing, initiated from points that are estimated based on the angles of the fitted lines and just outside the exclusion region; (d) the refined landmark location is estimated by fitting lines denoted $L'(t_1)$, $L'(t_2)$, and $L'(t^*)$ and q' finding the point that is closest to these lines.

which is where a ray from the previous landmark location in the previous blood vessel direction intersects the boundary of the exclusion region. For each trace, these new refined trace points and the previous landmark location are used to fit lines and recompute the landmark location. The previous landmark location is added to ensure stability, especially for intersections with acute angles.

6.8.1.5 Simplified Estimation of Landmark Model Parameters

The most expensive (and unstable) part of the estimation process is back-trace refinement. It makes sense to consider the possibility of a simpler algorithm where back-trace refinement is removed from the iterative estimation process

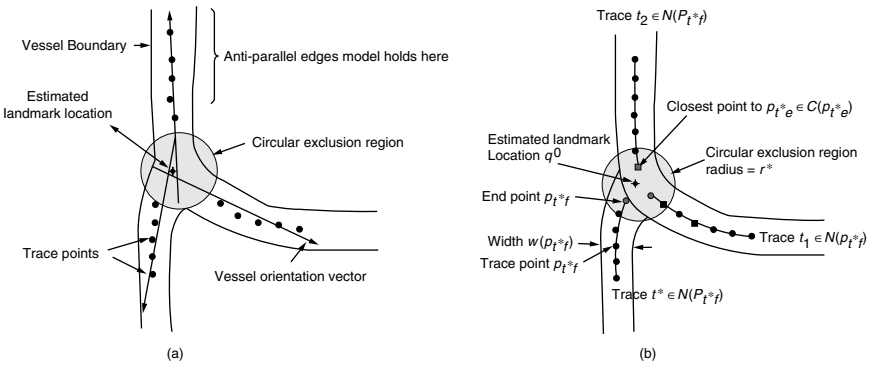


FIGURE 6.25 The landmark (intersection) model. Panel a shows the terminology. The vessels are assumed to exhibit pairs of anti-parallel edges/boundaries far away from intersections. The circular exclusion region is the region over which the anti-parallel edges model for vessels is not valid. It is also the region where the proposed model is valid and useful. Traces farther away from the exclusion region provide estimates of the local vessel orientations. The estimated landmark location is indicated by a star symbol. Panel b shows the mathematical notation.

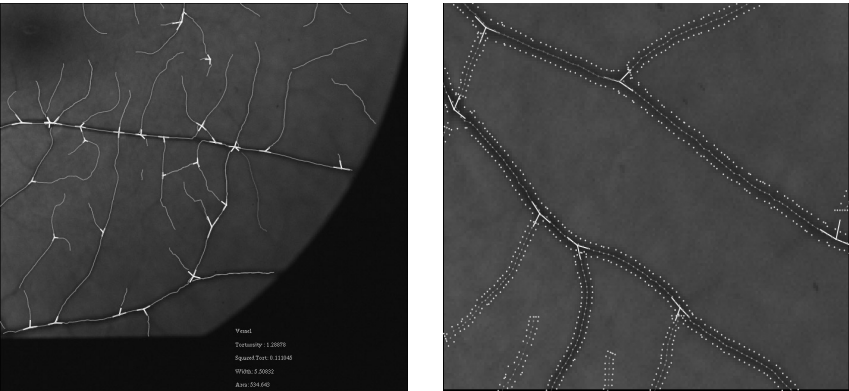


FIGURE 6.26 (See color insert.) The left panel shows how for a selected vessel, the vessel’s morphometric measures (tortuosity, width, area, etc.) can be calculated and displayed. The right panel shows the boundary locations of vessels that are needed to generate these measures.

and only applied at the end. In this simplified process, each iteration contains only two steps: estimation of the lines $L^j(t)$ for each trace in the landmark, followed by estimating q^j from these lines. After the iterative process, a single full iteration is applied, including back-trace refinement, line estimation, and centerline point estimation.

Using the same test criterion as above, if the new location is too far from the previous one, the landmark location is restored to its previous value.

Intuitively, this should work because it should be possible to obtain a reasonably accurate estimate of the landmark location from the traces outside the exclusion region, and allowing final accuracy of the centerline positions, trace orientations, and then landmark position in just a single, full iteration.

6.9 Applications of Vessel Segmentation Data

This section concerns the applications of retinal vessel tracing/segmentation that require specific structural and/or topological measurements to be generated from the images [107]. Typically, these measurements are needed in support of a clinical or biological hypothesis. The focus of these studies is often on quantifying changes effected by agents such as drugs, disease states, aging, development, and clinical treatments.

Commonly sought measurements include vessel widths and lengths [34, 83, 127], tortuosity [50, 52], and angles of intersection at crossing and branch locations. These measurements could be sought separately for arteries and veins, and different parts of the vasculature [56]. The thickness and length measurements are valuable measurements. The boundary length measurements are indicative of the surface area of the retinal vasculature. Means and variances on these quantities are also of common interest. Topological measurements usually involve the study of the above measures as a function of branching order.

Interestingly, most of these measurements are also used in the neurobiology community. Well-implemented software packages are commonly available for generating these measurements. The Neurolucida system (Microbrightfield, Inc., Colchester, VT, www.microbrightfield.com) is a good example of such a package [47, 64]. The file formats used by such packages are widely published. All that remains to leverage these tools is to write out the vessel tracing data into the format accepted by them.

6.10 Implementation Methods

Both hardware and software techniques have been used for retinal vessel tracing. Hardware techniques are useful for pixel processing algorithms when the application of interest requires fast computation, usually for real-time medical vision systems. With the development of fast exploratory algorithms, especially algorithms for intelligent scheduling of the tracing computations, and the rapid improvements in conventional processor technology, the need for specialized hardware is decreasing. A pure software-based approach also offers high levels of flexibility and portability across computing platforms.

Finally, even in pure software implementations, it is now possible to exploit processor features that are specifically aimed at image computing. For example, the Intel Pentium series of computers have a built-in instruction set, known as the MMX Multimedia Extension [13], that enables rapid computation of operations such as correlations. Often, the speedup attained by MMX is a factor of 2 to 3 over a generic implementation.

The modified version of Can's exploratory tracing algorithm described in Section 6.5.2.2 has been implemented for both a command-line and graphical user interfaces (GUIs). Known as RPI-Trace, it is written using a public-domain, open-source image-processing library known as VXL [118] that enables the source code to be compiled under multiple compilers in both Unix-based and Windows-based systems for a variety of commonly used graphical environments (GL, VTK, etc.) with no modifications.

The command-line version of the algorithm is useful for incorporation into an Internet application server or for batch processing of images. Users can upload images to a server, set desired parameters, and immediately download the results of automatic tracing.

Figure 6.27 shows a screen view of the graphical version of this implementation. The graphical user interface (GUI) allows a user to load a gray-scale or color image and, if desired, modify the image by applying local intensity normalization or selecting a specific color channel. The parameters can also be changed using provided menus if parameters other than the default are desired. Once the desired modification is made to the image, the exploratory tracing algorithm can be run in a piecemeal fashion or in its entirety, depending on the menu options selected. Results for each step (seed detection, seed verification, centerline trace identification, landmark determination, landmark refinement) of the algorithm can be toggled on and off and be viewed separately or collectively. The results can be saved in ASCII format for further analysis and can be again loaded directly into the GUI at a future time. Users also have the option of saving selected results as images in multiple formats. The algorithm can be run multiple times for the same or different images, and the results can be viewed by paging up and down through the results. This lends itself to batch processing of a set of images for different parameter settings using the command-line executable. Then, using the GUI, a user can load and view the results for all images in a single batch or for the same image across all the batches and page through the results. Also included in the GUI is the ability to zoom in and out, pan, and display the parameters used to generate the results with the option to make the parameters a permanent part of the results image.

Also included in the GUI is the ability for a user to manually designate vessels as vein, artery, or unknown. As each vessel segment is selected, morphological metrics such as tortuosity, median width, length, and area are computed and displayed. As the segmentation progresses, these metrics are combined to form cumulative statistics for areas and widths for each type of vessel.

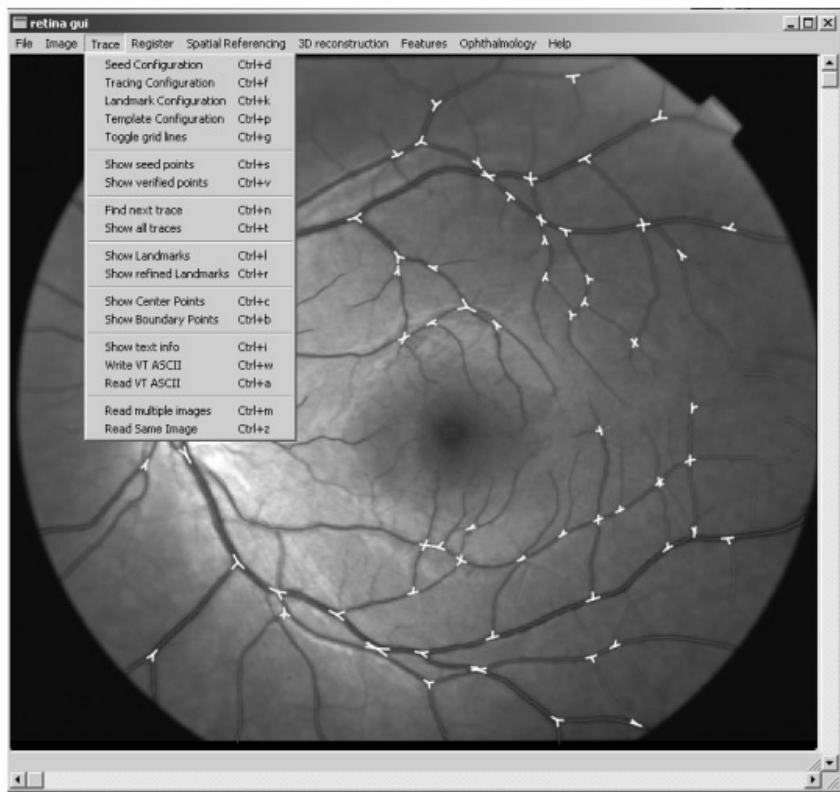


FIGURE 6.27 (See color insert.) Screen view of the RPI-Trace multi-platform graphical user interface. Shown are the currently options available for tracing vessels.

6.11 Experimental Validation Using Ground Truth Data

Validating the performance of automatic vessel segmentation algorithms is not straightforward, primarily due to the difficulty in identifying the “ground truth,” that is, establishing what is the “correct answer” or what *exactly* a computer segmentation is expected to produce. A secondary issue is the fact that, based on the application, the degree or amount of error that is acceptable varies and a way to quantify the error needs to be developed. For example, topological applications for studies of the vasculature may place a high emphasis on detection of all vascular segments or precision in determining the vessel boundaries, possibly to sub-pixel accuracy. On the other hand, applications that require vessel segmentation for registration purposes may have the

opposite emphasis. To validate a segmentation algorithm, the ground truth, the definition of what is being compared, and a measure are all required [78].

The arrival at “ground truth” is a known hard problem in image analysis and pattern recognition systems [54]. With retinal images, the ground truth is simply unavailable, and can be approximated by the creation of a “gold standard” to which computer-generated results are measured. In the context of this chapter, we define a gold standard as a binary segmentation, denoted as G , where each pixel, denoted as $G(x, y)$, assumes a value of 0 or 1 for background or vessel, respectively. With such a standard, a computer-generated segmentation, denoted C , can be compared and evaluated against the gold standard G . In evaluating each pixel, there are four possible cases. The case when $C(x, y) = G(x, y) = 0$ is called a “true negative.” The case when $C(x, y) = G(x, y) = 1$ is called a “true positive.” The case when $C(x, y) = 1$ and $G(x, y) = 0$ is called a “false positive.” Finally, the case when $C(x, y) = 0$ and $G(x, y) = 1$ is called a “false negative.” The frequency of these cases provides data that can be used as an indication of an algorithm’s performance.

Generation of gold standards is often a costly and time-consuming process. In addition, it is known that even expert human observers are subjective and prone to a variety of errors. For example, it is possible for one observer to label a vessel that another missed, and vice versa. This inconsistency is referred to as inter-observer variability. Likewise, the same human observer is very likely to generate different segmentations for the same image, which is known as intra-observer variability. Thus, the use of a single human expert’s annotations is unreliable and should be considered inadequate for the purpose of generating a gold standard. Thus, one approach to the creation of a gold standard is to combine multiple human-generated manual segmentations. From a set of multiple observers’ manual segmentations H , with each individual segmentation being denoted H_i , we wish to obtain a single binary segmentation G that will be considered the gold standard. We know that these segmentations will differ and thus a strategy for resolution of these differences must be created.

6.11.1 Ground Truth from Conflicting Observers

To determine the best way to combine different observer’s segmentation results, one must first consider what constitutes a correct and incorrect vessel segmentation. Essentially, there are three possible ways to define correct vessel segmentation for a particular pixel. A conservative method is to declare pixels to be part of a vessel (or part of the background) when all observers’ segmentations agree. If such total agreement exists, then that pixel would be marked as vessel (or background) in the gold standard image. In essence, this amounts to computing a Boolean “AND” of the multiple observer segmentations. A less conservative method would be based on using majority rule. Each segmentation result would contribute a single vote for each pixel when

determining if a pixel should be considered a vessel in the gold standard. If 50% or more of the observers have determined a particular pixel to be a vessel, it would be marked a vessel in the gold standard. Note that the “majority” threshold could be set higher or lower as appropriate for the specific application. The least conservative method is to label a pixel in the gold standard as a vessel pixel if it is marked as a vessel in at least one of the observers’ results. This last case is equivalent to computing the Boolean “OR” of the multiple observer segmentations.

An alternate approach to the gold standard generation process described above is to generate a modified gold standard in which each pixel is assigned a weight based on the number of observers who segmented that pixel as a vessel. This weight can be considered a probability as described in the next section. Figure 6.28 illustrates the resulting weights or probabilities of a non-binary gold standard formed from five separate hand-traced images.

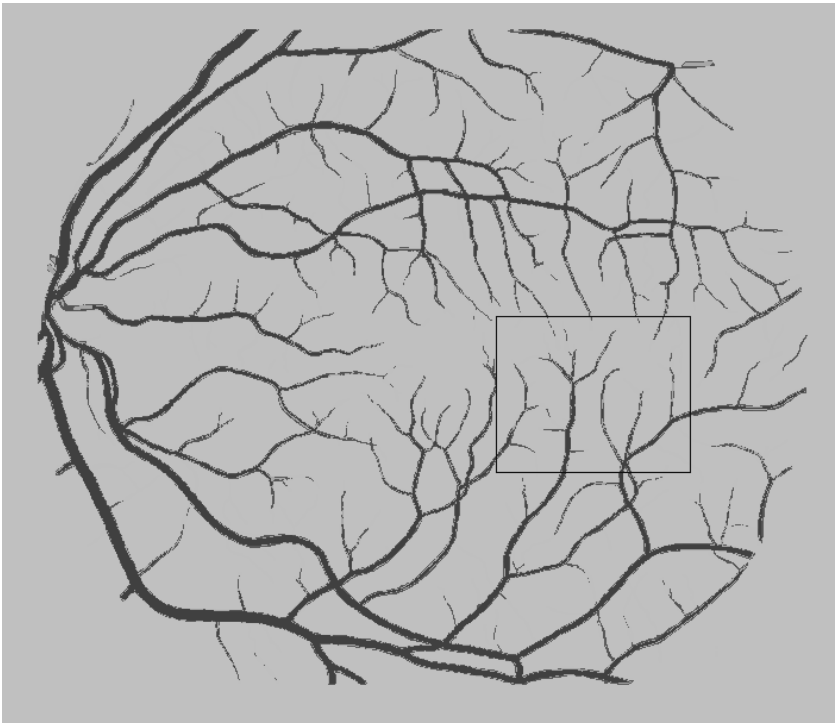


FIGURE 6.28 (See color insert.)

Multi-observer standard image formed from five separate hand tracings (two of the hand tracings courtesy Prof. Adam Hoover). Blue pixels are cases where five observers agree that the pixel is vessel; cyan is four; green is three; red is two; yellow is one; and white is zero.



FIGURE 6.29
Enlarged portion of the boxed region in [Figure 6.28](#) with the same color scheme described in [Figure 6.28](#) caption.

6.11.2 Multi-observer (Probabilistic) Gold Standard

Suppose that K is the number of observers ($K = 5$ in [Figure 6.28](#)). The segmentation label at each pixel for a computer-generated segmentation $C(x, y)$ can be thought of as a binary random variable V from the set $\{0, 1\}$. A value of $V = 0$ indicates a background pixel, and $V = 1$ indicates a vessel pixel. Let M denote an integer-valued random variable assuming values from the set $\{0, 1, \dots, K\}$, indicating the number of observers who have labeled a pixel as a vessel. The joint probability distribution $P_{VM}(v, m)$ can be estimated from the computer-generated segmentation and the manually scored segmentations. This distribution forms the basis for a performance measure S_{vessel} described below.

S_{vessel} was designed so that it assumes values in the interval from 0 to 1, where 1 indicates a perfect score and 0 a complete failure. A perfect score would indicate that all pixels determined by R or more observers to be vessel pixels have also been determined to be vessel pixels in C . A complete failure occurs when all vessel pixels determined by R or more observers are not labeled as vessel pixels in C . It is somewhere between 0 and 1, based on the number of pixels the computer does not label as vessel that are labeled as vessel by R or more observers. Note that in this method, described by [Equation 6.15](#), a missed vessel pixel in C , for which *all* observers agree to be

a vessel pixel, carries more weight in determining the final score than a pixel for which only R observers agree. R can be set to any appropriate value and we illustrate a “majority rules” idea in Figure 6.28 by choosing R such that it represents a majority (50% or higher) of the observers ($R = \lceil K/2 \rceil$). Based on these considerations, we define the S_{vessel} as follows:

$$S_{vessel} = \frac{E[M/V = 1; M \geq R]P[V = 1]}{E[M/V = 1, M \geq R]P[V = 1] + E[M/V = 0, M \geq R]P[V = 0]} \quad (6.15)$$

This simplifies to:

$$S_{vessel} = \frac{\sum_{m \geq R}^K m P[V = 1, M = m]}{\sum_{m \geq R}^K m P[M = m]} \quad (6.16)$$

In Equation 6.16, the numerator denotes the expected value of M for all the pixels for which the computer algorithm and R or more observers label as a vessel and can be thought of as the amount of agreement between C and the majority of observers. The denominator represents the average value of M for all the pixels that R or more observers have labeled as a vessel and can be thought of as the “perfect score.”

However, S_{vessel} does not take into account another indicator of segmentation performance, namely, the number of false positives, which in our context is the number of pixels falsely identified as vessels. The following second measure accounts for this aspect of segmentation performance:

$$F_{vessel} = \frac{P[V = 1, M = 0]}{P[V = 1, M \geq R]} \quad (6.17)$$

The numerator is the joint probability that the computer algorithm labeled the pixel to be a vessel and no observers labeled it as a vessel. The denominator is the probability that the computer algorithm labeled it as a vessel, and more than R observers agreed. In this measure, a score of 0 means that there were no false positives. A score of 1 would indicate that an equal number of false positives and true positives was required to achieve the S_{vessel} score.

A special case exists when there is only one observer. In this case, $K = R = 1$, and S_{vessel} and F_{vessel} become:

$$S_{vessel} = \frac{P[V = 1, M = 1]}{P[M = 1]} \quad (6.18)$$

$$F_{vessel} = \frac{P[V = 1, M = 0]}{P[V = 1, M = 1]} \quad (6.19)$$

Using the same type of reasoning, it is possible to define similar metrics, denoted $S_{background}$ and $F_{background}$, respectively, for classification of non-vessel pixels, as shown below:

$$S_{background} = \frac{E[M/V = 0, M < R]P(V = 0)}{E[M/V = 0, M < R]P(V = 0) + E[M/V = 1, M < R]P(V = 1)} \quad (6.20)$$

and

$$F_{background} = \frac{P(V = 0, M = K)}{P(V = 0, M < R)}. \quad (6.21)$$

Finally, the above metrics $S_{vessels}$ and $S_{background}$ could be combined simply by averaging to develop an overall score between 0 and 1. The closer a set of the results is to the multi-observer standard, the closer this score will be to 1. Figure 6.30 illustrates these performance measures.

6.11.3 Tracing

While the above measures would work for algorithms for which the goal is the segmentation of the entire vessel from the background, they would not work for algorithms in which the goal is to determine the centerlines of the vessels. For these algorithms, the following approaches can be used. The first approach is to convert a trace result into a segmentation result. This is often possible because most tracing algorithms determine the necessary data, such as vessel width or boundary location for each point found on the trace. Thus, a simple approach would be to segment as vessel points all the corresponding pixels between neighboring trace points based on the widths or vessel boundaries. This segmentation can then be scored as described above.

An alternative approach would be to test each point in the centerline trace against the multi-observer standard. Typically, exploratory algorithms generate results containing a set of center points separated by a fixed or varying step size. Another simple approach to judge the results of a tracing algorithm would be to test for true positives by seeing if a point in the centerline trace matches any non-zero (i.e., vessel) pixel in the multi-observer standard. If it does, it could be scored toward the S_{vessel} measure as described above or simply counted. Likewise, false positives could be scored or counted, and these combined measures could be used to compute measures indicating the algorithm's performance. However, such an approach fails to consider true and false positives because it is debatable as to what exactly constitutes a true negative or false negative. Thus, such an approach should not be considered; an approach that considers true and false negatives is discussed next.

A final approach is to build a trace based upon the multi-observer standard. This trace could be generated by running a tracing algorithm on the

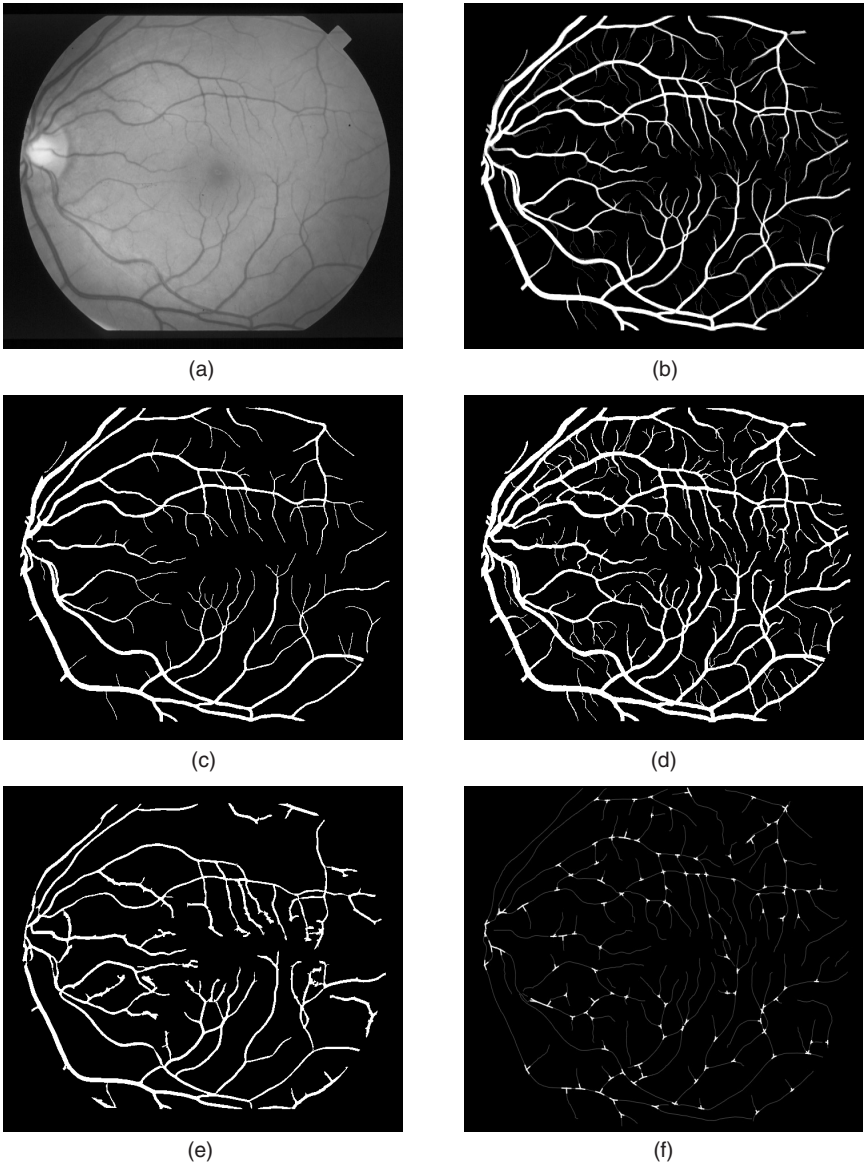


FIGURE 6.30

The measurement of segmentation performance from multiple manual observers. Panel a is the original image. Panel b is the multi-observer gold standard generated from five hand-traced images. Two of these manual traces are shown in panels c and d. Note the significant inter-observer variation. Panel e is a segmentation produced by Hoover's algorithm [53]. For this result, $S_{vessel} = 0.69$, $F_{vessel} = 0.14$, $S_{background} = 0.98$, and $F_{background} = 0.01$. Panel f is a tracing generated by RPI-Trace [21]. This tracing is accompanied by vessel boundary data that is not displayed here. For this result, $S_{vessel} = 0.94$, $F_{vessel} = 0.35$, $S_{background} = 0.94$, and $F_{background} = 0.002$. Based on these scores, the latter algorithm correctly detected more vessel pixels, but with a higher incidence of false positives.

multi-observer standard and then manually editing the results to generate the standard with which to compare tracing results. Once the tracing standard has been determined, false/true positive and false/true negative counts can be determined using a distance tolerance d . A true positive is a trace point in the image to be scored that is within distance d of a trace point in the tracing standard. A false positive is a trace point in the image to be scored that has no corresponding point in the tracing standard within distance d . A false negative would be a trace point in the tracing standard that has no corresponding trace point in the image to be scored within distance d . All others are true negatives.

6.11.4 Synthetic Images

Another avenue to consider when trying to gauge the performance of a vessel segmentation or tracing algorithm is the use of synthetic images. The advantage to such an approach is that ground truth can be absolutely known beforehand. However, in the context of retinal images, no matter how much care and attention is applied in trying to create such a synthetic image, it will never be a true recreation of an actual retinal image and, depending on how constructed, can be argued to favor one model over another. Thus, as a means of comparing algorithms, the use of synthetic images should not be considered. However, they may prove to be useful for a researcher to validate or measure the accuracy of a particular model or empirically test sensitivity of a particular model to factors such as noise or irregular illumination. In such cases, a researcher may wish to test an algorithm's performance with varying degrees of noise, or test the accuracies of morphometric measures such as width or tortuosity.

6.12 Experimental Analysis of Model and Settings for RPI-Trace

This section provides the reader with some insight into the effect the selection of models and parameters has on tracing performance. The description below focuses on RPI-Trace, which is based on the exploratory tracing algorithms described in Section 6.5.2.2. The main adjustable parameters include the density of initial sampling using vertical and horizontal cross sections (grid lines), tracing step size, and the expected maximum width of vessels.

Increasing the number of vertical and horizontal cross sections improves the probability of detecting vessels at the expense of computation. [Figure 6.31](#) shows the linear relationship between the number of detected vessel points and the grid lines. Depicted on this graph are the number of vessel points detected by the one- and two-dimensional parallel edge models, both using the

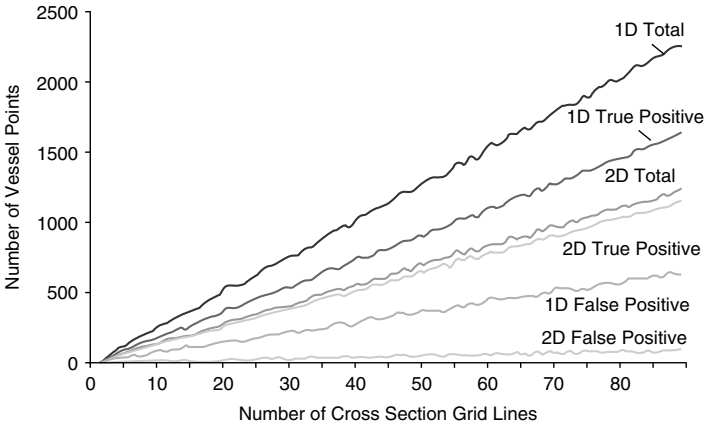


FIGURE 6.31

Graph depicting the results of varying the number of vertical and horizontal cross sections on the one-dimensional parallel edge and two-dimensional parallel edge models described in Section 6.4.6. Both models used local thresholds and the corresponding images with the results are seen in Figure 6.8 and Figure 6.10. This illustrates that there is a trade-off in using the two-dimensional model, in that the cost of eliminating a large percentage of the one-dimensional false positives results in a loss of true positives.

local minima and edge strength constraints (i.e., local thresholds) described in Section 6.4.6. The number of true positives and false positives, which are found based on comparison against the multi-observer standard (described in Section 6.11) are also depicted. The number of two-dimensional total positives is less than the number of one-dimensional true positives. This implies that the two-dimensional model is more tightly constrained to favor filtering out the false positives at the cost of filtering out some true positives. Notice that the ratio of one-dimensional false positives to true positives is relatively high when compared to the ratio of two-dimensional false positive to true positives.

The graph in Figure 6.32 demonstrates improved detection of vessels, as measured by the number of detected centerline points (trace points), as a function of the grid lines. Also shown are the true positive and false positive counts, again determined by comparison with the multi-observer standard. The rapid increase for a small number of grid lines (less than 5) can be mostly attributed to the thick and long vessels. The subsequent slower increase and leveling off is mostly due to the secondary and tertiary vessels. Figure 6.32, which plots the number of feature points (landmarks) detected as a function of grid size exhibits a similar increase. This can be attributed to the corresponding increase in the number of trace points. As more trace points are found in an image, more trace points are found in the vicinity of landmarks, thus allowing for their detection.

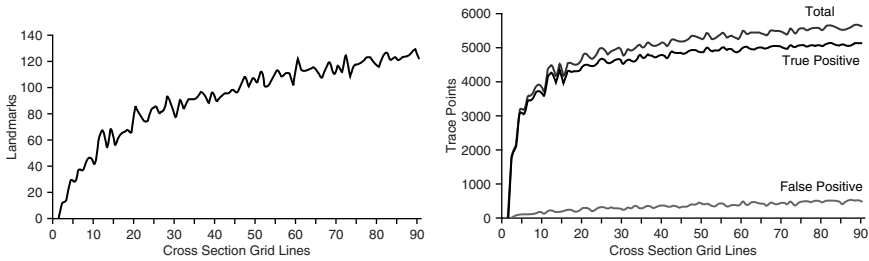


FIGURE 6.32

The graph on the left shows how the number of trace points detected varies as a function of the number of horizontal and vertical cross sections (grid lines). Also shown is the effect the number of grid lines has on the number of true positives and false positives. The graph on the right shows how the number of grid lines affects the number of detected landmarks.

The step size (α in Equation 6.8) is another important settable parameter. Small values of α result in more precise tracing at the expense of more computation. Generally, for a given density of initial sampling grid lines, the computation time decreases linearly [21] with increasing step size α .

6.13 Experimental Assessment of the Impact of Landmark Refinement with the ERPR Algorithm

This section describes experiments measuring the quantitative impact of using the ERPR landmark refinement algorithm described in Section 6.8. The method for measuring the accuracy and repeatability of landmark estimation is based on the image registration algorithm developed in our prior work [22, 23] as a testbed and standard for measuring landmark estimation errors. The average alignment error of this registration algorithm, as measured by the distance between trace centerlines is 0.83 pixels on 1024×1024 images.

The ability to register retinal images with sub-pixel accuracy, *despite errors in estimating the landmark locations (albeit at a high computational price)*, leads to several methods of evaluating the proposed Exclusion Region and Position Refinement (ERPR) algorithm.

- **Repeatability measurement.** For a landmark that appears in two or more images, registering the images places the two different estimated positions in the same coordinate system. This gives us a measure of the repeatability of the landmark position, modulo the transformation error itself.
- **Accuracy measurement.** We can compare the SSD refined positions obtained just prior to convergence to the estimated landmark positions as a further measure of position accuracy.

- We can generate a qualitative, visual indication of the effectiveness of the ERPR technique by transforming many different images of the same landmark into the same coordinate system. Examples of this are shown in [Figure 6.23](#).

The first quantitative measure is the error between the same landmark estimated in registered pairs of images. Let q_1 be the landmark location in image I_1 and q_2 the landmark location in image I_2 . Let θ be the estimated registration function mapping I_1 onto I_2 , so that $M(q_1;\theta)$ is the mapping of the landmark location into image I_2 . The local distance $\|M(q_1;\theta) - q_2\|$ gives one measure of the landmark error. The maximum difference in trace orientations between the mapping of the landmark at q_1 and the landmark at q_2 gives a second measure. By combining these measures over all landmarks in all registered image pairs, we obtain summary statistics on the repeatability of the estimated landmark parameters.

Table 6.1 shows these summary statistics for the original landmark detection technique and for several versions of our new ERPR technique, including the full method, the simplified method using one iteration of backtrace refinement, and a version where back-trace refinement is not used at all. Several conclusions are immediately apparent. First, the ERPR method is twice as repeatable as the original method, both in terms of position and orientation. In fact, it may be higher because of the inherent error in the transformation. Second, there is little difference between the fully iterative ERPR and the simplified version where back-trace refinement is used only once. Third, with no use of back-trace refinement, the results are substantially worse. Based on the latter two conclusions, the remaining experiments will focus exclusively on the ERPR with one step of back-trace refinement.

The second quantitative comparison is between the SSD-refined landmark positions that emerge at the end of registration and the positions estimated by either the original landmark technique or by the ERPR. Using corresponding point locations q_1 and q_2 , as above, the SSD step during estimation of θ refines

TABLE 6.1
Landmark Repeatability Experiment

	Location Distance		Max. Orientation Difference	
	Median (pixels)	Mean (pixels)	Median (degrees)	Mean (degrees)
Original	2.0879	2.2117	7.2037	8.6345
ERPR	1.0546	1.3408	3.7587	4.8014
ERPR, simplified	1.0713	1.3317	4.3324	5.4378
ERPR, no back-trace	1.2765	1.5126	5.0371	6.0380

The original Landmark Method is compared to three versions of the new ERPR Method: the full iterative refinement, the simplified version with one step of back-trace refinement, and an even simpler version with no back-trace refinement whatsoever.

TABLE 6.2
Landmark Position vs. SSD Refined Position

	Location Error	
	Median (pixels)	Mean (pixels)
Original	2.04	1.76
ERPR, simplified	1.10	0.72

Comparing the original landmark estimation technique to the new ERPR method with a just one step of back-trace refinement.

the position of q_2 to match the transformation of a small region surrounding q_1 . Call the new position q'_2 . The error measure is then $\|q_2 - q'_2\|$. Interestingly, this measure is not sensitive to minor errors in the transformation, and therefore an even better measure of the repeatability of the landmark position estimate. The disadvantage is that the SSD refinement gives no orientation accuracy.

The results of this measure taken over all correctly registered pairs from our data set are shown in Table 6.2. Both median and average errors are given. The advantage of the new ERPR method with just a single iteration of back-trace refinement is striking. The average error is 2.5 times lower and the median error is 1.9 times lower.

Overall, the ERPR algorithm reduces the median location error from 2.04 pixels down to 1.1 pixels, while improving the median spread (a measure of repeatability) from 2.09 pixels down to 1.05 pixels. Errors in estimating vessel orientations were similarly reduced from 7.2° down to 3.8° . These improvements are especially significant for real-time image registration applications for which computationally expensive refinement approaches such as sum of squared difference (SSD) registration can be avoided.

6.14 Chapter Summary

This chapter has surveyed some of the published vessel segmentation and tracing algorithms in the specific context of retinal fundus images. While many of the models and algorithms are influenced by related work in other areas of human anatomy, such as cardiac and brain images [110], and dye-injected neuron images [2], the techniques described in this chapter have been developed specifically in the retinal image context. We have attempted to describe some of the considerations of the many models and algorithms in use, in the context of the desired applications.

Among the applications, we broadly recognize two classes: real-time and online vessel tracing applications, and offline applications. In our experience, the online applications are best served by exploratory approaches, while the offline applications are best served by a combination of filtering and exploratory tracing approaches. For the latter, a detailed graph-based output such as the Neurolucida file format [47] provides the most flexibility overall.

We recognize that our classification of algorithms, as reflected in this chapter, is only one of several ways of organizing the body of reported work. Our own work, principally represented by exploratory algorithms, is heavily influenced by the desire to process live image sequences in real-time. It is also influenced heavily by the desire to use conventional off-the-shelf computers rather than specialized hardware whenever possible. Yet another influence on our work is the desire to maintain a common software code base for real-time and offline tracing modules. The former are being incorporated into an ophthalmic instrument under construction. The latter are being adapted for use on an Internet application server. This server enables potential users, especially clinicians, to upload images via the Internet and then download the results of vessel tracing, registration [22], and mosaicing [23].

Much work remains in this field. For example, automatic algorithms for classifying vessels as arteries and veins are still unavailable. Also not available are algorithms that automatically handle normal as well as the “hollow” vessels noted earlier in a reliable manner using multiple models. Finally, there is a continuing need to improve the computational efficiency. For example, the refinement of landmark locations (Section 6.8) is still relatively slow.

More generally, vessel-tracing algorithms have the potential to form the basis for a complete retinal image understanding system. Notwithstanding some prior work in this area [50], much work remains. In the area of real-time tracing, the desire to attain ever higher frame rates (200 frames/sec is considered an appropriate goal given the speed of eye motions [88, 94]). Higher frame rates imply noisier images. Much work remains to be done on handling noisy retinal images in an efficient manner, while using more sophisticated/realistic models of the vasculature that account for the imaging noise, and the varied appearance of the vasculature.

Acknowledgments

Various portions of this research were supported by the National Science Foundation Experimental Partnerships grant EIA-0000417, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award Number

EEC-9986821), the National Institutes for Health grant RR14038, and by Rensselaer Polytechnic Institute.

The authors would like to thank the staff at the Center for Sight, especially photographers Gary Howe and Mark Fish, for assisting with image acquisition. Thanks to Dr. Anna Majerovics for valuable guidance on retinal structure and function, and for numerous images. Thanks also to Matthew Freshman for a part of the retinal photography. Thanks to Dr. Wayne Whitwam and Jay Cohn at the University of Minnesota Medical Center for images and insights related to hypertensive retinopathy. Thanks to Professor Adam Hoover at Clemson University for sharing the manually traced image data, and permission to use this data in this chapter. The authors appreciate the insightful input of Dr. George Nagy at Rensselaer, especially regarding vectorization methods from the document image analysis literature.

Finally, thanks to the other associated members of the RPI Retina Project group, including Chris Carothers, Amitha Perera, Gary Yang, Gang Lin, Mohamad Tehrani, Justin LaPre, and Michal Sofka, for their input, ideas, support, artwork, and friendship.

References

1. Akita, K. and Kuga, H., "A Computer Method of Understanding Ocular Fundus Images," *Pattern Recognition*, Vol. 15, No. 6, pp. 431–443, 1982.
2. Al-Kofahi, K., Lasek, S., Szarowski, D. H., Pace, C., Nagy, G., Turner, J. N., and Roysam B., "Rapid Automated Three-Dimensional Tracing of Neurons from Confocal Image Stacks," *IEEE Trans. Information Technology in Biomedicine*, accepted 2002.
3. Anarim, E., Aydinoglu, H., and Goknar, I. C., "Decision Based Edge Detector," *Signal Processing*, Vol. 35, pp. 149–156, 1994.
4. Balduf, E., "Real-Time Algorithms for a Laser Retinal Surgery System—Implementation on the Silicon Graphics Octane Dual-Processor System," M. S. thesis, Rensselaer Polytechnic Institute, 1998.
5. Barrett, S. F., Jerath, M. R., Rylander, H. G., and Welch, A. J., "Digital Tracking and Control of Retinal Images," *Optical Engineering*, Vol. 33, No. 1, pp. 150–159, 1994.
6. Barrett, S., Wright C., and Jerath, M., "Computer-Aided Retinal Photocoagulation System," *J. Biomed. Opt.*, Vol. 1, pp. 83–91, 1996.
7. Beach, M., Tiedeman, J. S., Hopkins, M., and Sabharwal, Y., "Multi-spectral Fundus Imaging for Early Detection of Diabetic Retinopathy," *Proc. SPIE Conf. Clinical Diagnostic Systems and Technologies*, Vol. 3603, pp. 114–121, 1999.
8. Becker, D., "Algorithms for Automatic Retinal Mapping and Real-Time Location Determination for an Improved Retinal Laser Surgery System," Ph. D. thesis, Rensselaer Polytechnic Institute, August 1995.

9. Becker, D., Can, A., Tanenbaum, H., Turner, J., and Roysam, B., "Image Processing Algorithms for Retinal Montage Synthesis, Mapping, and Real-Time Location Determination," *IEEE Trans. Biomed. Eng.*, Vol. 45, No. 1, 1998.
10. Berger J., Patel T., Shin D., Piltz J., and Stone R., "Computerized Stereo Chronoscopy and Alternation Flicker to Detect Optic Nerve Head Contour Change," *Ophthalmology*, Vol. 107, No. 7, pp. 1316–1320, 2000.
11. Berger, J. and Shin D., "Computer Vision Enabled Augmented Reality Fundus Biomicroscopy," *Ophthalmology*, Vol. 106, No. 10, 1999.
12. Beucher S., "Watershed, Hierarchical Segmentation and Waterfall Algorithm," *Mathematical Morphology and Its Applications to Image Processing*, Serra, J. and Soille, P., Eds., Kluwer Academic, pp. 69–76, 1994.
13. Bistry, D., Ed., *The Complete Guide to MMX Technology*, Intel Corp. and McGraw-Hill Publishing, 1997.
14. Borodkin, M. and Thompson, J., "Retinal Cartography: An Analysis of Two-Dimensional and Three-Dimensional Mapping of the Retina," *Retina — The J. Retinal and Vitreous Diseases*, Vol. 12, No. 3, pp. 273–280, 1992.
15. Boudier, H., Le Noble, J., Messing, M., Huijberts, M., Le Noble, F., and Van Essen, H., "The Microcirculation and Hypertension," *J. Hypertension*, Vol. 10, pp. S147–S156, 1992.
16. Brinchmann-Hansen, O. and Heier, H., "Theoretical Relations between Light Streak Characteristics and Optical Properties of Retinal Vessels," *Acta Ophthalmol.*, pp. 33–37, 1986.
17. Brown, L., "A Survey of Image Registration Techniques," *ACM Computing Surveys*, Vol. 24, No. 4, pp. 325–376, 1992.
18. Can, A., "Robust Computer Vision Algorithms for Registering Images from Curved Human Retina," Ph.D. thesis, Rensselaer Polytechnic Institute, 2000.
19. Can, A., Stewart, C., and Roysam, B., "Robust Hierarchical Algorithm for Constructing a Mosaic from Images of the Curved Human Retina," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999.
20. Can, A., Stewart, C., Roysam, B., and Tanenbaum, H., "A Feature-Based Technique for Joint, Linear Estimation of High-Order Image-to-Mosaic Transformations: Application to Mosaicing the Curved Human Retina," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000.
21. Can, A., Shen, H., Turner, J., Tanenbaum, H., and Roysam, B., "Rapid Automated Tracing and Feature Extraction from Live High-Resolution Retinal Fundus Images Using Direct Exploratory Algorithms," *IEEE Trans. Information Technology in Biomedicine*, Vol. 3, No. 2, pp. 125–138, 1999.
22. Can, A., Stewart, C., Roysam, B., and Tanenbaum, H., "A Feature-Based Robust Hierarchical Algorithm for Registration Pairs of Images of the Curved Human Retina," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 3, 2002.
23. Can, A., Stewart, C., Roysam, B., and Tanenbaum, H., "A Feature-Based Algorithm for Joint, Linear Estimation of High-Order Image-to-Mosaic Transformations: Mosaicing the Curved Human Retina," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 3, 2002.
24. Canny, J., "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679–698, 1986.

25. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., and Goldbaum, M., "Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters," *IEEE Trans. Medical Imaging*, Vol. 8, No. 3, pp. 263–269, 1989.
26. Chutatape, O., Zheng, L., and Krishnan, S., "Retinal Blood Vessel Detection and Tracking by Matched Gaussian and Kalman Filters," *Proc. IEEE Int. Conf. Engineering in Medicine and Biology Society*, Vol. 20, No. 6, pp. 3144–3149, 1998.
27. Cideciyan, A., "Registration of Ocular Fundus Images," *IEEE Engineering in Medicine and Biology*, Vol. 14, No. 1, pp. 52–58, 1995.
28. Clark, T., Freeman, W., and Goldbaum, M., "Digital Overlay of Fluorescein Angiograms and Fundus Images for Treatment of Subretinal Neovascularization," *Retina—J. Retinal and Vitreous Diseases*, Vol. 2, No. 12, pp. 118–126, 1992.
29. Coatrieux, J., Garreau, M., Collorec, R., and Roux, C., "Computer Vision Approaches for the Three-Dimensional Reconstruction: Review and Prospects," *Crit. Rev. Biomed. Eng.*, Vol. 22, No. 1, pp. 1–38, 1994.
30. Cohen, A., Roysam, B., and Turner, J., "Automated Tracing and Volume Measurements of Neurons from 3-D Confocal Fluorescence Microscopy Data," *J. Microsc.*, Vol. 173, Pt. 2, 1994.
31. Collorec, R. and Coatrieux, J. L., "Vectorial Tracking and Directed Contour Finder for Vascular Network in Digital Subtraction Angiography," *Pattern Recog. Lett.*, Vol. 8, No. 5, pp. 353–358, Dec. 1988.
32. Cooper, J., Venkatesh, S., and Kitchen, L., "Early Jump-out Corner Detectors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 8, pp. 823–829, 1993.
33. Dani, P. and Chaudhuri, S., "Automated Assembling of Images—Image Montage Preparation," *Pattern Recognition*, Vol. 28, No. 1, pp. 431–445, 1995.
34. Delori F., Fitch K., Feke G., Deupree D., and Weiter, J., "Evaluation of Micrometric and Microdensitometric Methods for Measuring the Width of Retinal Vessel Images on Fundus Photographs," *Graefe's Arch. Clin. Exp. Ophthalmol.*, Vol. 226, pp. 393–399, 1988.
35. Denninghoff, K. and Smith, M., "Optical Model of the Blood in Large Retinal Vessels," *J. Biomed. Opt.*, Vol. 5, No. 4, pp. 371–374, 2000.
36. Dimmitt, S., West, J., Eames, S., Gibsin, J., Gosling, P., and Littler, W., "Usefulness of Ophthalmoscopy in Mild to Moderate Hypertension," *Lancet*, pp. 1103–1106, 1989.
37. Dougherty, E. R., *An Introduction to Morphological Image Processing*, Soc. Photo-Optical and Instrumentation. Eng., Bellingham, WA, 1992.
38. Eichel, P., Delp, E., Koral, K., and Buda, A., "A Method for a Fully Automatic Definition of Coronary Arterial Edges from Cineangiograms," *IEEE Trans. Med. Imag.*, Vol. 7, pp. 313–320, 1988.
39. Family, F., Masters, B., and Platt, D., "Fractal Pattern Formation in Human Retinal Vessels," *Physica D*, Vol. 38, pp. 98–103, 1989.
40. Federman, J. L., Ed., *Retina and Vitreous*. The C. V. Mosby Company, St. Louis, 1988.
41. Fine, S., "Observations Following Laser Treatment for Choroidal Neovascularization," *Arch. Ophthalmol.*, Vol. 106, pp. 1524–1525, 1988.
42. Flower, R. and Hochheimer, B., "A Clinical Technique and Apparatus for Simultaneous Angiography of the Separate Retinal and Choroidal Circulation," *Investigative Ophthalmol.*, Vol. 12, No. 4, pp. 248–261, 1973.

43. Francois, D. and Djemel, Z., "Extracting Line Junctions from Curvilinear Structures," *Proc. Geosci. Remote Sensing Symp., IGARSS, IEEE 200 International*, Vol. 4, pp. 1672-1674, 2000.
44. Gang, L., Chutatape, O., and Krishnan, S., "Detection and Measurement of Retinal Vessels in Fundus Images Using Amplitude Modified Second-Order Gaussian Filter," *IEEE Trans. Biomed. Eng.*, Vol. 49, pp. 168-172, 2002.
45. Gao, X., Bharath, A., Stanton, A., Hughes, A., Chapman, N., and Thom, S., "A Method of Vessel Tracking for Vessel Diameter Measurement on Retinal Images," *IEEE Int. Conf. Image Processing*, pp. 881-884, 2001.
46. Garreau, M., Coatrieux, J., Collorec, R., and Chardenon, C., "A Knowledge-Based Approach for 3-D Reconstruction and Labeling of Vascular Networks from Bi-plane Angiographic Projections," *IEEE Trans. Med. Imag.*, Vol. 10, No. 2, pp. 122-131, 1991.
47. Glaser J. R. and Glaser E., Neuron Imaging with Neurolucida. A PC-Based System for Image Combining Microscopy, *Computerized Medical Imaging and Graphics*, Vol. 14, pp. 307-317, 1990.
48. Goldbaum, M., Kouznetsova, V., Cot, B., Hart, W., and Nelson, M., "Automated Registration of Digital Ocular Fundus Images for Comparison of Lesions," *SPIE: Ophthalmic Technologies III*, Vol. 1877, pp. 94-99, 1993.
49. Goldbaum, M., Katz, N., Chaudhuri, S., Nelson, M., and Kube, P., "Digital Image Processing for Ocular Fundus Images," *Ophthalmol. Clinics N. Am.*, Vol. 3, No. 3, pp. 447-466, 1990.
50. Goldbaum, M., Moezzi, S., Taylor, A., Chatterjee, S., Boyd, J., Huner, E., and Jain, R., "Automated Diagnosis and Image Understanding with Object Extraction, Object Classification, and Inferencing in Retinal Images," *1996 Proc. IEEE Int. Conf. on Image Processing*, Vol. 3, pp. 695-698, 1996.
51. Grewal, M. and Andrews, A., *Kalman Filtering: Theory and Practice*, Prentice Hall, 1993.
52. Hart, W., Goldbaum, M., Cote, B., Kube, P., and Nelson, M., "Automated Measurement of Retinal Vascular Tortuosity," *Proc. AMIA Fall Conf.*, pp. 459-463, 1997.
53. Hoover, A., Kouznetsova, V., and Goldbaum, M., "Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response," *IEEE Trans. Med. Imag.*, Vol. 19, No. 3, pp. 203-210, 2000.
54. Hu, J., Kahsi, R., Lopresti, D., Nagy, G., and Wilfong, G., "Why Table Ground-Truthing Is Hard," *Proc. Sixth Int. Conf. Document Analysis and Recognition*, pp. 129-133, 2001.
55. Hubbard, L., Brothers, R., King, W., Clegg, L., Klein, R., Cooper, L., Sharrett, A., Davis, M., and Cai, J., "Methods for Evaluation of Retinal Microvascular Abnormalities Associated with Hypertension/Sclerosis in the Atherosclerosis Risk in Communities Study," *Ophthalmology*, Vol. 106, No. 12, pp. 2269-2280, 1999.
56. Jagoe, R., Arnold, J., Blauth, C., Smith, P., Taylor, P., and Wootton, R., "Measurement of Capillary Dropout in Retinal Angiograms by Computerized Image Analysis," *Patt. Recog. Lett.*, Vol. 13, pp. 143-151, 1992.
57. Janssen, R. and Vossepoel, A., "Adaptive Vectorization of Line Drawing Images," *Computer Vision and Image Understanding*, Vol. 65, No. 1, pp. 38-56, 1997.
58. Jasiobedzki, P., Williams, C., and Lu, F., "Detecting and Reconstructing Vascular Trees in Retinal Images," *Medical Imaging 1994: Image Processing*, Murray H. Loew, Ed., Vol. 2167, pp. 815-825, 1994.

59. Kitchen, L. and Rosenfeld, A., "Gray Level Corner Detection," *Patt. Recog. Lett.*, Vol. 1, pp. 95–102, 1982.
60. Klein, A., Egglin, T., Pollak, J., Lee, F., and Amini, A., "Identifying Vascular Features with Orientation Specific Filters and B-spline Snakes," *Computers in Cardiol.*, pp. 113–116, 1994.
61. Klein, R., Klein, B., Neider, M., Hubbard, L., Meuer, S., and Brothers, B., "Diabetic Retinopathy as Detected Using Ophthalmoscopy, a Non-mydratic Camera and a Standard Fundus Camera," *Ophthalmology*, Vol. 92 No. 4, pp. 485–491, 1985.
62. Kochner, B., Schuhmann, D., Michaelis, M., Mann, G., and Englmeier, K.-H., "Course Tracking and Contour Extraction of Retinal Vessels from Color Fundus Photographs: Most Efficient Use of Steerable Filters for Model-Based Image Analysis," *Proc. SPIE*, Vol. 3338, *Medical Imaging 1998: Image Processing*, Kenneth M. Hanson, Ed., pp. 755–761, 1998.
63. Kyriacos, S., Nekka, F., and Cartilier, L., "Insights into the Formation Process of the Retinal Vasculature," *Fractals*, Vol. 5, No. 4, pp. 615–624, 1997.
64. Lambe, E., Krimer, L., and Goldman-Rakic, S., "Differential Postnatal Development of Catcholamine and Serotonin Inputs to Identified Neurons in Prefrontal Cortex of Rhesus Monkey," *J. Neurosci.*, Vol. 20, No. 23, pp. 8780–8787, 2000.
65. Laplante, P., *Real-Time Systems Design and Analysis, An Engineer's Handbook*, 2nd edition, IEEE Press, 1997.
66. Li, H. and Chutatape, O., "Automatic Location of the Optic Disk in Retinal Images," *Proc. IEEE Int. Conf. Image Processing*, pp. 837–840, 2001.
67. Li, H. and Chutatape, O., "Fundus Image Features Extraction," *Proc. Annu. Eng. Bio. Soc., (EMBS) Int. Conf.*, pp. 3071–3073, 2000.
68. Lu, S. and Eiho, S., "Automatic Detection of the Coronary Arterial Contours with Sub-branches from an X-ray Angiogram," *Computers in Cardiol.*, pp. 575–578, 1993.
69. Mahurkar, A., Trus, B., Vivino, M., Kuehl, E., Datiles, M., and Kaiser-Kupfer, M., "Retinal Fundus Photo Montages: A New Computer Based Method," *Investigative Ophthalmol. Visual Sci.*, Vol. 36, No. 4, 1995.
70. Mahurkar, A., Vivino, M., Trus, B., Kuehl, E., Datiles, M., and Kaiser-Kupfer, M., "Constructing Retinal Fundus Photomontages," *Investigative Ophthalmol. Visual Sci.*, Vol. 7, No. 8, pp. 1675–1683, 1996.
71. Markow, M., Rylander, H., and Welch, A., "Real-Time Algorithm for Retinal Tracking," *IEEE Trans. Biomed. Eng.*, Vol. 40, No. 12, pp. 1269–1281, 1993.
72. Mokhtarian, F. and Suomela, R., "Robust Image Corner Detection Through Curvature Scale Space," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. 20, No. 12, pp. 1376–1382, 1998.
73. Monahan, P., Gitter, K., Eichler, J., Cohen, G., and Schomaker, K., "Use of Digitized Fluorescein Angiogram System to Evaluate Laser Treatment for Subretinal Neovascularization: Technique," *Retina — J. Retinal and Vitreous Diseases*, Vol. 13, No. 3, pp. 187–195, 1993.
74. Murphy, R., "Age-Related Macular Degeneration," *Ophthalmology*, Vol. 93, pp. 969–971, 1986.
75. Natarajan, S., *Imprecise and Approximate Computation*, Kluwer Academic, Boston, 1995.
76. Nayar, S., Baker, S., and Murase, H., "Parametric Feature Detection," *Proc. IEEE Conf. Computer Vision and Patt. Recog.*, pp. 471–478, 1996.

77. Neumann, F., Schreiner, W., and Neumann, M., "Computer Simulation of Coronary Arterial Trees," *Adv. Eng. Software*, Vol. 28, pp. 353–357, 1997.
78. Niessen, W., Bouma, C., Vincken, K., and Viergever, M., "Error Metrics for Quantitative Evaluation of Medical Image Segmentation," *Performance Characterization in Computer Vision*, Klette, R., Stiehl, H., Viergever, M., and Vincken, K., Eds., Kluwer Academic, Dordrecht, The Netherlands, pp. 275–284, 2000.
79. Nguyen, T. and Sklansky, J., "Computing the Skeleton of Coronary Arteries in Cineangiograms," *Comput. Biomed. Res.*, Vol. 19, pp. 428–444, 1986.
80. Onuki, T. and Nitta, S., "Computer Simulation of Geometry and Hemodynamics of Canine Pulmonary Arteries," *Ann. Biomed. Eng.*, Vol. 21, pp. 107–115, 1993.
81. Otsu, N., "A Threshold Selection Method from Grey-Level Histograms," *IEEE Trans. Syst., Man, and Cybernet.*, Vol. 9, pp. 62–66, 1979.
82. Parida, L., Geiger, D., and Hummel, R., "Junctions: Detection, Classification, and Reconstruction," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. 20, No. 7, pp. 687–698, 1998.
83. Pedersen, L., Grunkin, M., Ersboll, B., Madsen, K., Larsen, M., Christoffersen, N., and Skands U., "Quantitative Measurement of Changes in Retinal Vessel Diameter in Ocular Fundus Images," *Patt. Recog. Lett.*, Vol. 21, pp. 1215–1223, 2000.
84. Peli, E., Augliere, R., and Timberlake, G., "Feature-Based Registration of Retinal Images," *IEEE Trans. Med. Imag.*, Vol. 6, No. 3, 1987.
85. Pinz, A., Bernogger, S., Datlinger, P., and Kruger, A., "Mapping of the Human Retina," *IEEE Trans. Med. Imag.*, Vol. 17, No. 4, pp. 606–619, 1998.
86. Polli, R. and Valli, G., "An Algorithm for Real-Time Vessel Enhancement and Detection," *Comput. Meth. Programs in Biomed.*, Vol. 52, pp. 1–22, 1997.
87. Press, W., Flannery, B., Tenkolsky, S., and Vetterling, W., *Numerical Recipes in C — The Art of Scientific Computing*, Cambridge University Press, Cambridge, England, 1990.
88. Rayner, K., Ed., *Eye Movements and Visual Cognition: Scene Perception and Reading*, Springer Series in Neuropsychology, Springer-Verlag, New York, 1992.
89. Roberts, D., "Analysis of Vessel Absorption Profiles in Retinal Oximetry," *Medical Phys.*, Vol. 14, pp. 124–130, 1987.
90. Rohr, K., "Recognizing Corners by Fitting Parametric Models," *Int. J. Comput. Vision*, Vol. 9, No. 3, pp. 213–230, 1992.
91. Sato, Y., Nakajima, S., Nobuyuki, S., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., and Kikinis, R., "Three-Dimensional Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images," *Med. Image Anal.*, Vol. 2, No. 2, pp. 143–168, 1998.
92. Sebok, T., Roemer, L., and Malindzak, G., Jr., "An Algorithm for Line Intersection Identification," *Patt. Recog.*, Vol. 13, No. 2, pp. 159–166, 1981.
93. Shen, H., "Optical Instrumentation and Real-Time Image Processing Algorithms for Simultaneous ICG and Red-Free Video Angiography of the Retina," M.S. thesis, Rensselaer Polytechnic Institute, 1996.
94. Shen, H., "Indexing Based Frame-Rate Spatial Referencing Algorithms: Application to Laser Retinal Surgery," Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY, 2000.
95. Shen, H., Lin, G., Stewart, C. V., Tanenbaum, H. L., and Roysam, B., "Frame-Rate Spatial Referencing Based on Invariant Indexing and Alignment with

- Application to Laser Retinal Surgery," *Proc. IEEE Computer Society Conf. Computer Vision and Patt. Recog.*, Kauai, Hawaii, 2001.
96. Shen, H., Roysam, B., Stewart, C.V., Turner, J.N., and Tanenbaum, H.L., "Optimal Scheduling of Tracing Computations for Real-time Vascular Landmark Extraction from Retinal Fundus Images," *IEEE Trans. Information Technol. Biomed.*, Vol. 5, No. 1, 2001.
 97. Sherman, T., "On Connecting Large Vessels to Small," *J. Gen. Physiol.*, Vol. 78, pp. 431, 1981.
 98. Simoncelli, E. and Farid, H., "Steerable Wedge Filters for Local Orientation Analysis," *IEEE Trans. Image Processing*, Vol. 5, No. 9, pp. 1377–1383, 1996.
 99. Sinthanayothin, C., Boyce, J., Cook, H., and Williamson, T., "Automated Localisation of the Optic Disk, Fovea, and Retinal Blood Vessels from Digital Colour Fundus Images," *Br. J. Ophthalmol.*, Vol. 83, No. 8, 1999.
 100. Smith, M., Denninghoff, K., Lompado, A., and Hillman, L., "Effect of Multiple Light Paths on Retinal Vessel Oximetry," *Appl. Opt.*, Vol. 39, No. 7, pp. 1183–1193, 2000.
 101. Sonka, M., Hlavac, V., and Boyle, R., *Image Processing, Analysis, and Machine Vision*, Brooks/Cole, Pacific Grove, CA, 1999.
 102. Sonka, M., Winniford, M., and Collins, S., "Coronary Borders in Complex Images," *IEEE Trans. Med. Imag.*, Vol. 14, No. 1, pp. 151–161, 1995.
 103. Sonka, M., Winniford, M., and Collins, S., "Reduction of Failure Rates in Automated Analysis of Difficult Images: Improved Simultaneous Detection of Left and Right Coronary Borders," *Comput. Cardiol.*, pp. 111–114, 1992.
 104. Staib, L. and Duncan, J., "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Patt. Anal. Machine Intelligence*, Vol. 14, No. 11, 1992.
 105. Stanton, A., Mullaney, P., Mee, F., O'Brien, E., and O'Malley, K., "A Method of Quantifying Retinal Microvascular Alterations Associated with Blood Pressure and Age," *J. Hypertension*, Vol. 13, No. 1, pp. 41–48, 1994.
 106. Stanton, A., Wasan, B., Cerutti, A., Ford, S., Marsh, R., Sever, P., Thom, S., and Houghes, A., "Vascular Network Changes in the Retina with Age and Hypertension," *J. Hypertension*, Vol. 13, pp. 1724–1728, 1995.
 107. Stokoe, N. and Turner R., "Normal Retinal Vascular Pattern Arteriovenous Ratio as a Measure of Arterial Calibre," *Br. J. Ophthalmol.*, Vol. 50, No. 21, pp. 21–40, 1966.
 108. Stromland, K., Hellstrom, A., and Gustavsson, T., "Morphometry of the Optic Nerve and Retinal Vessels in Children by Computer-Assisted Image Analysis of Fundus Photographs," *Graefe's Arch. Clin. Exp. Ophthalmol.*, Vol. 233, pp. 150–153, 1995.
 109. Sun, Y., "Automated Identification of Vessel Contours in Coronary Arteriograms by an Adaptive Tracking Algorithm," *IEEE Trans. Med. Imag.*, Vol. 8, pp. 78–88, 1989.
 110. Sun, Y., Lucariello, R., and Chiaramida, S., "Directional Low-Pass Filtering for Improved Accuracy and Reproducibility of Stenosis Quantification in Coronary Arteriograms," *IEEE Trans. Med. Imag.*, Vol. 14, No. 2, 1995.
 111. Tan, W., Wang, Y., and Lee, S., "Retinal Blood Vessel Detection Using Frequency Analysis and Local-Mean-Interpolation Filters," *Medical Imaging 2001: Image Processing*, Milan Sonka and Kenneth M. Hanson, Eds., Vol. 4322, pp. 1373–1384, 2001.

112. Toledo, R., Radeva, P., Von Land, C., and Villanueva, J., "3D Dynamic Model of the Coronary Tree," *IEEE Computers in Cardiol.*, Vol. 25, p. 777, 1998.
113. Trokel, S. L., "Lasers in Ophthalmology," *Optics and Photonics News*, pp. 11–13, 1992.
114. Tso, M. and Jampol, L. M., "Pathophysiology of Hypertensive Retinopathy," *Ophthalmologica*, Vol. 89, p. 1132, 1982.
115. Turner, J., Shain, W., Szarowski, D., Lasek, S., Sipple, B., Pace, C., Al-Kofahi, K., Can, A., and Roysam, B., "Confocal Light Microscopy of Brain Cells and Tissue: Image Analysis and Quantitation," *Acta Histochem. Cytochem.*, Vol. 32, No. 1, pp. 5–11, 1999.
116. Tyler, M. and Saine, P., *Ophthalmic Photography: Retinal Photography, Angiography, and Electronic Imaging*, 2nd edition, Butterworth-Heinemann Medical, 2002.
117. Van Cuyck, P., Gerbrands, J., and Reiber, J., "Automated Centerline Tracing in Coronary Angiograms," *Pattern Recognition Artificial Intelligence*, pp. 169–183, 1998.
118. VXL, *The VXL Book*, <http://vxl.sourceforge.net/>, 2002.
119. Wang, Y. and Lee, S., "A Fast Method for Automated Detection of Blood Vessels in Retinal Images," *Conf. Record of the Thirty-First Asilomar Conf. on Signals, Systems and Computers*, Vol. 2, pp. 1700–1704, 1997.
120. Watson, G. S., *Statistics on Spheres*, John Wiley & Sons, New York, 1983.
121. Wink, O., Niessen, W., and Viergever, M., "Fast Delineation of Vessels in 3-D Angiographic Images," *IEEE Trans. Med. Imag.*, Vol. 19, No. 4, pp. 337–346, 2000.
122. Wright, C., Ferguson, R., Rylander H., III, Welch, A., and Barrett, S., "Hybrid Approach to Retinal Tracking and Laser Aiming for Photocoagulation," *J. Biomed. Opt.*, Vol. 2, No. 2, pp. 195–203, 1997.
123. Yannuzzi, L., Guyer, D., and Green, R., *The Retina Atlas* (on CD-ROM), Mosby Year Book, Inc., St. Louis, MO, 1998.
124. Zana, F. and Klein, J.-C., "A Multimodal Registration Algorithm of Eye Fundus Images Using Vessels Detection and Hough Transform," *IEEE Trans. Med. Imag.*, Vol. 18, No. 5, pp. 419–428, 1999.
125. Zana, F. and Klein, J.-C., "Robust Segmentation of Vessels from Retinal Angiography," *Proc. Int. Conf. Digital Signal Processing*, pp. 1087–1091, 1997.
126. Zana, F. and Klein, J.-C., "Segmentation of Vessel-like Patterns Using Mathematical Morphology and Curvature Evaluation," *IEEE Trans. Image Processing*, Vol. 10, No. 7, pp. 1010–1019, 2001.
127. Zhou, L., Rzeszutarski, M., Singerman, L., and Chokreff, J., "The Detection and Quantification of Retinopathy Using Digital Angiograms," *IEEE Trans. Med. Imag.*, Vol. 13, No. 4, pp. 619–626, 1994.
128. Martin, D. F., Sierra-Madero J., Walmsley S., Wolitz R. A., Macey K., Georgiou P., Robinson C. A., and Stempien M. J., "A Controlled Trial of Valganciclovir as Induction Therapy for Cytomegalovirus Retinitis," *New Engl. J. Med.*, Vol. 346, No. 15, pp. 1119–1126, 2002.