Non-recursive paired tracking for vessel extraction from retinal images

Marc Lalonde†, Langis Gagnon† and Marie-Carole Boucher‡
†Centre de recherche informatique de Montréal
550 Sherbrooke W., Suite 100, Montréal, Qc, H3A 1B9
{mlalonde, lgagnon}@crim.ca
‡Département d'ophtalmologie, Hôpital Maisonneuve-Rosemont
5689 Boul Rosemont, Montréal, Qc¸ H1T 2H1

Abstract

This paper reports on a novel vessel tracking algorithm being developped as part of a set of tools for the automated diagnosis of diabetic retinopathy. The algorithm tracks vessels in low-resolution optical images of the retina using their edge map as computed by the Canny edge operator. Tracking proceeds by following an edge line (which coincides with a vein border) while monitoring the connectivity of its twin border. Breaks in the connectivity trigger the creation of seeds that serve as extra starting points for future tracking. Seed creation allows the algorithm to handle bifurcations (a key issue in vessel tracking) and jump over broken or missing edges. Results are presented for five typical fundus images.

Introduction

Vessel tracking in fundus images is of significant importance for many reasons. First, it provides a map of the retinal vessels of the eye, from which a reference frame may be derived that would ease the process of associating objects (e.g. lesions, anatomical structures) with absolute coordinates. Second, it allows the localization of stable anchor points (such as bifurcations) needed for image registration. Third, it allows the characterization of veins and thus the potential detection of a very specific pathology called veinous beading (tortuosities in veins that reflect the progress of diabetic retinopathy). Finally, it turns out to be helpful to other recognition algorithms by facilitating the removal of the vascular network. (For instance, algorithms dedicated to the detection of microanyeurisms are known to 'get confused' by the presence of vessels.)

A large collection of algorithmic tools and techniques have been employed to recover veins from greyscale images. Curiously, one avenue that has been barely explored is the traditional, computer-vision approach based on edge detection and linking. This paper reports on a new algorithm for vessel extraction from optical images of the retina, with the use of classical edge detection (i.e. the Canny operator) as a preprocessing stage. The algorithm is designed to satisfy these goals: First, it should isolate veins with good accuracy (in terms of localization) and efficiency (high recognition percentage, low false negatives, low false positives); second, bifurcations should be detected; and third, it should be relatively fast.

The paper is arranged as follows. Section 1 exposes a substantial survey of the literature on vessel extraction with emphasis on ophthalmology. Image acquisition is described in Section 2 along with some preprocessing steps before invoking the proposed algorithm detailed in Section 3. Some results are given and discussed in Section 4 and finally, future work is proposed.

1 State of the art

One well-known technique for vessel extraction is the matched-filter method introduced by Chaudhuri et al. [1] and further improved by Hoover et al. [6]. It consists of a set of 2D kernels that are swept across the grayscale image, where each kernel "models" a piece of vessel at a certain orientation (discretized at every 15 degrees) with a Gaussian cross-section and a given length. The output is an image with pixel values proportional to the best filter response. So the approach acts as a smart detector of parallel lines with Gaussian intensity profile in between. It is worth noting that compared to the methods reported here, the matched-filter technique possesses some interesting feactures, such as robustness, no initialization required and no user intervention needed. However it also suffers from a number

Presented at the Conference Vision Interface 2000, Montréal, Mai 2000

of limitations: it requires intensive computations due to the large kernels used, additional work is necessary to recover the actual vessels and the bifurcations from the output image, extra filtering is needed to reduce false alarms, and threshold selection may be critical. Hoover et al. went further by proposing a complementary method to obtain better separation: the filter output is locally analyzed and a local threshold is derived. The algorithm is reported to be capable of managing bifurcations while reducing the amount of false positive responses compared to Chaudhuri's version.

Some researchers have studied the color properties of fundus images. In most cases, segmentation techniques have been proposed that aim at enhancing retinal pathologies such as microaneurysms and hemorrhages (Donohoe et al. [2], Truitt et al. [14]) or cytomegalovirus retinitis (Fontaine et al. [4]), but a more pertinent application, reported in Rakotomalala et al. [11], includes color as a criteria for vessel tracking. They first perform color edge detection using a recursive Deriche filter (the CIEL*a*b* color space is reported to give the best results). Tracking is initiated near the optic disk by the selection of two starting points L_1 and R_1 as nodes, lying respectively on the left and right borders of the vessel to be extracted. The next pair of nodes $(L_{2,}R_2)$ in the tracking direction is then retrieved, and so on. Color is still used during tracking since 'colorimetric validation' ensures that the vein cross-section between the two current nodes found on the vessel borders by the tracking algorithm has similar color properties than that of the vessel body under construction. The algorithm is said to be robust to missing edge points, and also bifurcations are detected and explored recursively. The operator is asked to select the pair of starting points for each vessel that he wants to extract.

Kochner et al. [7] split the problem of vessel extraction in two stages. First, the centerline of each vessel is extracted with the help of steerable filters in a manner which is similar to the approach in [17], i.e. tracking of the centerline starts from a point close to the optic disk and progresses inside the main branch by selecting the next point that yields the strongest filter response. Second, the vessel contour is determined (again using filter response in a direction perpendicular to the centerline) and vessel branches are detected in the same process. Interestingly, starting points for tracking are determined automatically; the authors propose an initialization method that consists of forming a circle around the optic disk and retaining as starting points the edge points that intersect the circle. Obviously, the optic disk must be present in the image.

Tolias and Panas [13] use a similar technique for initialization but they tackle the problem of vessel identification within the framework of fuzzy logic. Tracking is based on fuzzy clustering of pixels belonging to a potential vessel. Two classes are defined: "vessel" and "non-vessel", and the membership function of the class "vessel" is made adaptive with respect to some statistics of the vessel profile (normal to the tracking direction) at each iteration. T-junctions and forks are detected following analysis of the vessel profile. Tracking terminates when vessel contrast along a profile is low or when more than two groups of pixels belong to the "vessel" class.

Zana and Klein [15] focused on the contribution of mathematical morphology for vessel identification. They basically rely on the same hypotheses found in [1], namely that the vessels are piecewise linear and that their cross-section is Gaussian-shaped, but they apply morphological operators to the (angio) image in order to enhance the vessels. The image is first opened with a set of linear structuring elements (15-pixel long, every 15^o) and then the small vessels (capillaries) are recovered after geodesic reconstruction. Additional operators are used to eliminate noise due for example to the fluorescence of the layers below the retina. The overall objective is to use the method as a first step for image registration (actually explained in [16]).

Additional algorithms have been developed for vessel extraction from angio images and are thus not quite adapted to the current problem for two reasons. First, the modality is different : angio images are monochrome while optical images contain color information, and many image properties such as resolution, noise content, contrast, etc. are different. Some researchers (e.g. Figueiredo and Leitao [3], Miles and Nuttall [10]) have modeled the intensity profile of vessels in angio images as a Gaussian shape and have used this model for vessel identification; so far we have not found any indication that this model is adequate for the analysis of low resolution optical images. Second, in some cases the objectives are also different. For example, Sun et al. [12] propose a technique for stenosis quantification, which obviously includes vessel identification, but since its dominant feature is the accuracy of the vessel border delineation, extra work would be required to adapt it to vessel extraction in ophthalmic images.

The most common problem undermining the effectiveness of many algorithms concerns the handling of bifurcations, in the sense that most may fail to detect them. For example, Lecornu et al. [8] see vessel tracking as an optimization problem where the two best paths between two vessel end points must be recovered (where the paths should obviously coincide with the vessel borders). The cost function to be minimized includes parameters related to edge contrast and parallelism but no provision is made to allow the detection and correct handling of bifurcations. Other tracking techniques (e.g. Zhou et al. [17]) rely on the output of a matched filter designed to match the 1D Gaussian intensity profile of the vessel (perpendicular to the tracking direction) and again, tracking may ignore side branches or fail upon reaching a bifurcation. However Liu and Sun [9], although using a matched filter, introduce the possibility of recursively managing branches and bifurcations.

2 Image preprocessing

2.1 Image acquisition

As said before, color images of the patient's retina are provided by a non-mydriatic, 3 CCD analog camera. The camera generates 24-bit, RGB images of 640 x 480 pixels in size, which approximately yields an approximate resolution of 20 μm per pixel. Typical vein diameter is about 100 μm near the optic disk (the optic nerve), which translates into a 4-5 pixel diameter, and can be as thin as 1 pixel.

2.2 Color preprocessing

Extensive use of color segmentation techniques was investigated in the preprocessing stage but was discarded for a number of reasons:

- The color distribution of retinal images may vary somewhat markedly from one patient to another (one factor being the patient's ethnic origin) because of the presence of pigments in the choroid and the epithelial layer of the retina. This would call for greater user intervention in selecting the appropriate vein color.
- Veins exhibit significant color variability depending on their localization. The appearance of large vessels is affected by the presence of 'backlight' caused by light reflections of the camera flash onto the optic disk. In fact, backlight reveals the three-dimensional arrangement of the vascular network: some veins clearly belong to different planes, and it impacts on the pixel intensity[1]. Moreover, smaller vein sections in peripheral areas of the retina do not get the same light exposure as those near the optic disk, so non-uniform illumination results from the fact that the formed image is a projection of a 3-D spheric scene (the retina) onto the 2D sensor plane.
- Color segmentation often implies imprecise object contour as a result of the color pixel classification process; region growing might be used to recover missing pixels but that might not be enough to provide smooth and accurate vein borders.

Although some work has been published lately in the area of retinal image segmentation (e.g. [14], [4]), additional studies are needed to assess the true performance of the proposed solutions. Paradoxically, simple color-to-greyscale transforms seem to provide good contrast for vein border detection and tracking, and the observation is in agreement with other researchers (e.g. Kochner et al. [7], Chaudhuri et al. [1]). The green band of the original RGB signal has been empirically selected as the greyscale representation that yields better contrast, although other transforms have also been considered (the intensity band of the HSI color space, the lightness band of the CIEL*a*b* color space).

2.3 Edge detection

Unlike many algorithms which rely on simple low-level edge detection (e.g. Sobel), edge detection is performed using the Canny operator. The choice is justified according to the following benefits provided by the operator :

- optimal edge placement: good localization (i.e. edge points as close as possible to the center of the true edge) was one of Canny's criteria in the design of his optimal step edge detector. In short, instead of developing an ad hoc border detection stage as part of the main tracking method we let the Canny operator do what it is good at.
- computation of an estimate of the normal vector to the edge: this estimate will be useful for tracking, as explained later in the next section.

Although one property of Canny's edge detection operator is the production of edges with 1-pixel width, a thinning pass is still required to make the edge map more appropriate for tracking with the removal of some extra edge pixels around junctions and along curved segments. After thinning (we used Guo & Hall's algorithm [5]), the edge map has the following properties : 1) any pixel in the middle of a line segment has two neighbors only; 2) an end pixel has one neighbor; 3) if a pixel has more than two neighbors, it belongs to a junction.

[1] In some images we have found that backlight is sometimes so strong that it saturates the CCD sensor.

3 Tracking

This section exposes the design of the vein tracking algorithm, starting with specifications that summarize the expected behavior of the algorithm and introducing its various components and mechanisms.

3.1 Specifications

A number of factors and expected properties (specifications) should guide the design of the tracking algorithm:

- the tracker should obviously exploit the 'parallel' nature of vessel borders;
- it should be tolerant to broken edges, i.e. 'jump' over gaps between edges and still keep tracking a vessel;
- it should be tolerant to noise, i.e. degrade gracefully when following a wrong track induced by edge detection artifacts;
- it should not be stopped by bifurcations (implicit property: it should detect them and keep tracking along their branches);
- it should extract most of the vascular network while minimizing user intervention.

3.2 Overview of the algorithm

The algorithm is designed to track the borders of a vessel sequentially, following one border at a time while registering the presence of the "twin" border, until one of the following events occurs: 1) the end of the line is reached, or 2) a junction is met (since the algorithm cannot privilege one direction, it stops).

As a consequence, a piece of vessel is analysed twice, and both analyses should be in agreement. The dual tracking process improves the quality of recognition and ensures that bifurcations are handled properly. The ability of the algorithm to jump over gaps and go beyond bifurcations is provided by a "seed" generation mechanism that analyses some properties of the "twin" points and creates new starting points (called seeds) for subsequent tracking according to rules such as connectivity between twin points.

3.3 Preliminary definitions

Tracking is performed in the edge map I(x,y) produced in the edge detection phase, but it also makes use of the orientation map Θ(x,y) computed by the Canny operator. The orientation map is actually filtered before its use. A simple averaging filter was selected to smooth out sharp angle variations. Each point in Θ(x,y) gives us an estimate of the orientation θ of the normal to the edge at (x,y). A tracking attempt is initiated from a triple (C$_0$, $\vec{p}_0$, dir) called a **seed**: C_0 is a point in I(x,y) called the origin (the starting point for tracking) and $\vec{p}_0$ sets the progress direction (in which direction should tracking proceed), while dir equals -1 or 1 and specifies which direction (left or right) to look at when searching for a twin point. Obviously, dir is a constant during tracking.

3.4 Locating a twin point

Since the edge map has been previously skeletonized, one may view any vessel border as a chain of edge pixels, where a pixel has a predecessor and a successor, depending on the tracking direction (provided that the pixel is not an endpoint or a junction point). Let us focus on a point C$_i$ somewhere along the border; its predecessor is denoted C$_{i-1}$, and both points define a progress vector $\vec{p}_{C_{i-1},C_i}$, or, more shortly, $\vec{p}_i$. A twin point to C$_i$, labeled T$_i$ and belonging to the twin border, is lying somewhere on the lefthand side or the righthand side of C$_i$ with respect to the tracking direction. At any point C$_i$ along the border, the search for T$_i$ takes place in the continuation of the normal to the curve $\overrightarrow{n_i}$ (whose orientation is given by θ_i in the Θ image). Figure 1 sketches these variables. Since two directions are possible for $\overrightarrow{n_i}$ (θ or $\theta - \pi$, depending on the computation of the orientation by the Canny operator), the direction is chosen according to the rule:

$$\angle\,\overrightarrow{n_i} = \begin{cases} \theta & \text{if}\,\mathrm{sign}(\overrightarrow{p_i} \times \overrightarrow{n_i}) \;=\; \mathrm{sign}(dir) \\ \theta - \pi & \text{otherwise} \end{cases}$$

In addition to that, domain knowledge about plausible vessel diameters helps deduce reasonable bounds in the search for T_i along $\vec{n}_i$ (the distance between C_i and T_i gives an estimate of the vessel diameter). Bounds are kept loose (e.g. 1-2 pixels for the lower bound; three times the average vessel diameter for the higher bound) so they should not influence performance in terms of quality of recognition.

In some occasions, no T_i is found for a given C_i, either because the opposite border is broken or because $\vec{n}_i$ points toward the interior of a branch (e.g. see Figure 1, right). Missing T_i's are one cause of seed generation because their occurence signals unusual behavior of the twin border such as a change of border at a bifurcation point.

3.5 Seed generation

The basic border tracking algorithm described so far is given two simple tasks:

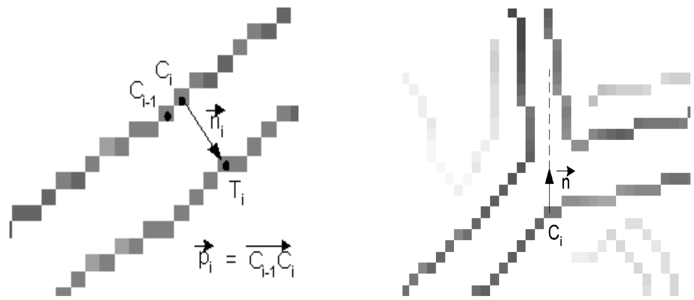

Figure 1: Left: definitions. Right: missing twin point due to the changing topology of the vein borders.

1. follow one border (starting from a given pair of points that determine the direction of progress) while registering the presence of its twin border.
2. detect unusual events during registration. Such events would notify the tracker about the fact that it is no longer registering the same twin border.

The power of the vein tracking algorithm lies in its ability to detect these events and create additional starting points on newly found borders. It effectively covers all borders of any bifurcation it encounters, regardless of its topology (e.g. number of branches). Seeds are created according to two sets of conditions: *conditions for creation* and *conditions for selection*. The conditions for creation are:

1. **First twin point of the sequence.** Right after tracking begins, the first twin point found induces a seed for future tracking. This rule allows both borders of a vein to be tracked "independently". In fact, two seeds are created: one for each possible direction.
2. **Broken connectivity of the T_i's**. During normal tracking mode, all twin points would belong to the same border and would thus be connected together; when T_i is found not to be connected to T_{i-1}, seeds must somehow be placed onto both pieces of borders.
3. **Missing T_i's.** Every time the search for a T_i fails, it also signals broken connectivity.

A direction of search (DoS) is also specified for each seed ; the DoS indicates where the twin border is expected to be found (either left or right) with respect to the direction of progress. Usually, such direction is opposite to the current DoS: if T_i lies at the right of C_i and it becomes a candidate for a seed origin, the new tracker should look at left for its own twin point (in the vicinity of C_i) if it tracks in the same direction, otherwise it should look at right. Algorithmically, the determination of the DoS follows this procedure:

- get the two neighbors to T_i, denoted by $N_i^{(1)}$ and $N_i^{(2)}$
- for each neighbor $N_i^{(j)}$, form a progress vector $\overrightarrow{p}_{T_i,N_i^{(j)}}$ and compute the dot product between it and the progress vector $\vec{p}_i$ defined earlier. If the result is positive, both vectors have the same orientation and their DoS should be opposite. If the dot product is negative, their DoS should be the same. Of course, we expect both seeds created at T_i to possess opposite DoS.

Once conditions for creation are met, the first appropriate T_i complying with the following *conditions for selection* is selected:

1. It should be preceded by a valid connected T_{i-1}; this condition makes the mechanism more robust by requiring some stability in the newly found piece of border. Otherwise, spurious seed points might be created;
2. It is required to have two neighbors, which will define two seed points;
3. Directions of search for both potential seeds with T_i as origin should be opposite;
4. T_i should not have been covered by a previous tracking event.

3.6 Additional considerations

In the overview of the algorithm given in Section 3.2, the list of stopping conditions does not include seed creation. Indeed, once tracking is initiated, it is not interrupted by the finding of a seed, as opposed to a recursive tracking algorithm that would reinitiate tracking each time a seed is found. We expect the non-recursive version to be more robust for the following reason: after completion of the tracking along a border, post-analysis of the result may recommend rejecting the candidate vessel on the basis of dubious attributes (e.g. unstable or diverging diameter estimate as tracking progresses) that would tend to indicate the finding of a false vessel. It would not make sense to retain and eventually make use of the seeds that might have been found during tracking of such a candidate vessel of poor quality.

It has been mentioned earlier (Section 3.2) that the tracker cannot handle junctions since it is not allowed to take a decision regarding which direction to pick. Recall that a junction is found during tracking when a point C_i has more than two neighbors. The strategy that has been adopted to solve the problem is a direct continuation of the seed mechanism: each branch of the junction is treated as a potential vein border and seeds are created accordingly. However a heuristic has been added to prevent the creation of multiple seeds in case a cluster of junctions happens to be in contact with the border being tracked.

Another topic related to junctions is the existence of 'hairs'. Noisy edge maps are known to contain hairy lines, that is, lines with very short edges perpendicular to them. Canny's non-maximum suppression stage helps minimize these by "inhibiting" edge pixels neighboring the dominant edges, but hairs still show up occasionally, breaking the momentum of the tracker (tracking stops at the junction between the hair and the border, and the hair is uselessly analysed during seed creation). So the creation of seeds on junction branches is made conditional to the length of the branches, and only those with minimal pixel length may receive seeds. Of course, in the special case of hairs, the only valid branch of the T-junction is the continuation of the border (the hair is ignored), and the tracker is simply allowed to pursue its course (no seeds are created).

3.7 Post-processing

Upon completion of vessel tracking based on the set of (provided) seeds, information about potential vessel borders is stored in memory. The post-processing stage aims at identifying the pairs of borders A_i and A'_i that 'match' together, that is, any point of A_i has its twin point on A'_i and, inversely, each point of A'_i has its twin point on A_i. These matching pairs of borders are believed to be pieces of vessels. Figure 2 illustrates the result of post-processing : the identified pairs are displayed and their centerline (i.e. the set of mid-points between points in a border and their twin points) is also shown. Note the existence of some invalid pairs (e.g. in the upper part of the image); it should normally be relatively easy to filter them out based on some criteria (e.g. connection with the rest of the network, valid and stable diameter, and so on). This stage prepares data for higher processing such as bifurcation / fork detection (not addressed in this paper).

4 Results and discussion

Five optical fundus images (labeled A, B, C, D and E) have been processed by the vein tracking algorithm described in the previous section. The images possess the characteristics described in Section 2.1 about image acquisition and they were acquired during examination of patients suffering from diabetic retinopathy.

The results after vessel extraction show the ability of the algorithm to successfully track and characterize not only the vessels that were 'selected' following the placement of the starting point but also nearby vessels that are somehow connected to them (e.g. via the optic disc). An eloquent example is shown for image A in Figure 2 (left), where tracking proceeded from two starting points. It is clear that the algorithm manages to create enough additional seeds by itself to cover much of the network. For example, in the case of image A, 618 seeds have been created during processing, 144 of which were found to be redundant, as explained in Section 3.5. Note that the image shows a rough estimate of the vascular network and that postprocessing is necessary.

Of course, a more complete solution (i.e. better network "coverage") would be obtained if additional starting points were supplied, and this brings up the question of how the starting points should be identified. The current method is simply through user interaction but an automatic procedure driven by one of the techniques exposed in Section 1 is planned : points lying on the main branches would be extracted with a method based on edge "inspection" around the optic disc (as advocated by [13],[7]). Additional strategies might also be considered such as using matched filters to locate parallel lines with high confidence and selecting line ends as seeds. The image B has been difficult to process due to its weak contrast. Although the edges of the main vessels are reasonably well defined, those of the smaller veins have much lower quality. Tracking obviously suffers from the presence of degraded edge structures, and that explains the result shown in Figure 2 (top right). Processing of the image C underlines the opposite issue. The edge map is so rich that tracking sometimes gets lost in the noise. Figure 2, bottom right, shows the tracking result, and one clearly sees the effect of noisy edges on vessel extraction. However, the right image tends to indicate that proper post-processing would still allow the recovery of the network itself while filtering noise with a procedure that would e.g. discard short pairs of borders with no neighbors in a reasonable range. Figure 3 shows results for the remaining images D and E.

The results reported in this section are qualitative. A more quantitative assessment of the performance of

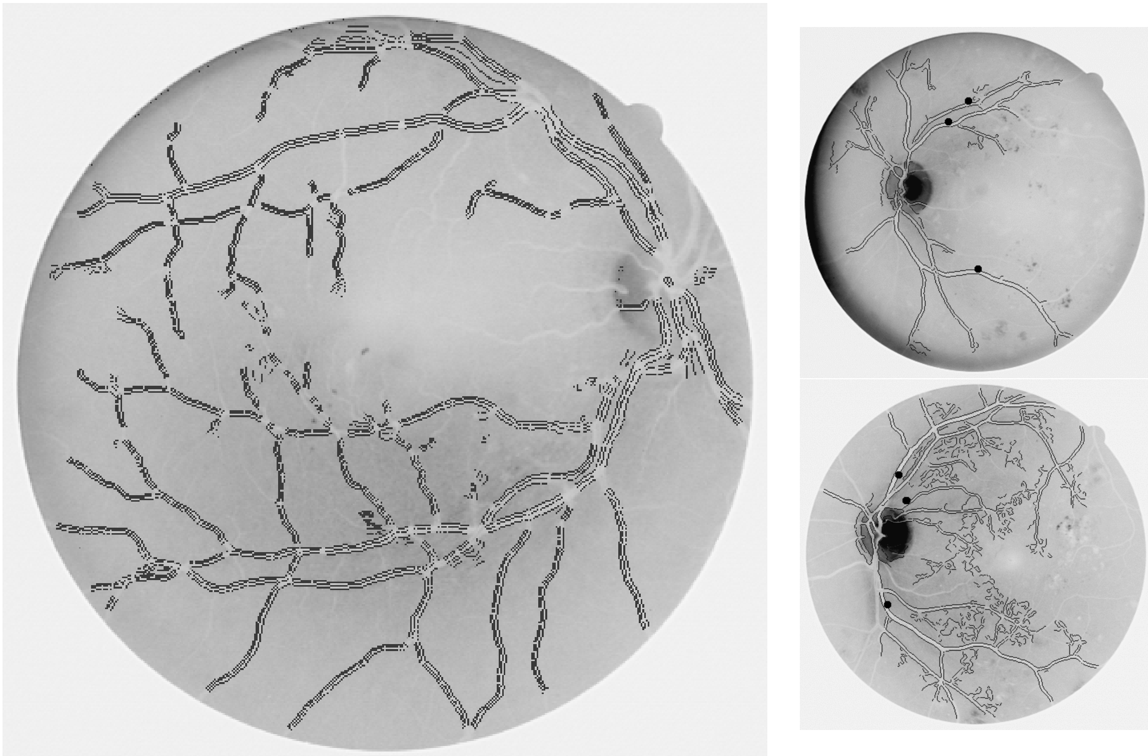

Figure 2: Left: the centerline of matching pairs of vein borders is shown. Right: tracking results for images B (top) and C (down), without preprocessing. The black dots represent the starting points supplied to the tracking algorithm.

the method has been found to be difficult to obtain since it would require a manual segmentation of the vessels by a "qualified" operator. This is a difficult and subjective task especially when it comes down to segmenting very small vessels or those that progressively fuse with the background. Segmenting the largest vessels may look simple but is in fact not trivial: how do we define the vessel borders ? where do we place the edges ? Future tests will be performed to quantify the "network coverage" by estimating the proportion of vessels whose center falls between pairs of detected borders (see Figure 2 for an illustration). These tests (and more especially the elaboration of an automated testbench) are important because they would allow us to validate design choices such as the selection of the greyscale representation (which one is best ?) and evaluate the sensitivity of thresholds and parameters (e.g. low and high thresholds of the edge detection stage).

As for speed performance, the time required to extract the vessel network from a typical image (using two starting points) is about 5 seconds on a 350 MHz Pentium II running Solaris 2.7. Preprocessing (edge detection followed by thinning and smoothing) takes roughly 3 seconds, while tracking itself obviously depends on the image contents (richness of the edge map) and the number of starting points given to the algorithm.

Conclusion and future work

A vessel tracking algorithm has been presented. Its design was based on many considerations, namely good accuracy and efficiency, appropriate handling of bifurcations, speed, tolerance to broken edges, tolerance to noise (graceful degradation in performance) and minimal user intervention. It has been found that the proposed algorithm complies with these features except the tolerance to noise which is very dependent upon the performance of the edge detection stage. Two aspects need to be explored: a better threshold setting is required for the Canny algorithm, and a "self-regulated" mechanism should be added to the tracker in order to bring it to a halt whenever a bad vein border candidate (or a noisy edge) is being followed. We are currently investigating the use of the evidential reasoning theory for such a mechanism, i.e. the local confidence level at any point during tracking would be a function of the geometry of the pair of borders being tracked; parallel borders should provide high confidence while a 'random walk' in clutter should lead to a fast-dropping confidence that

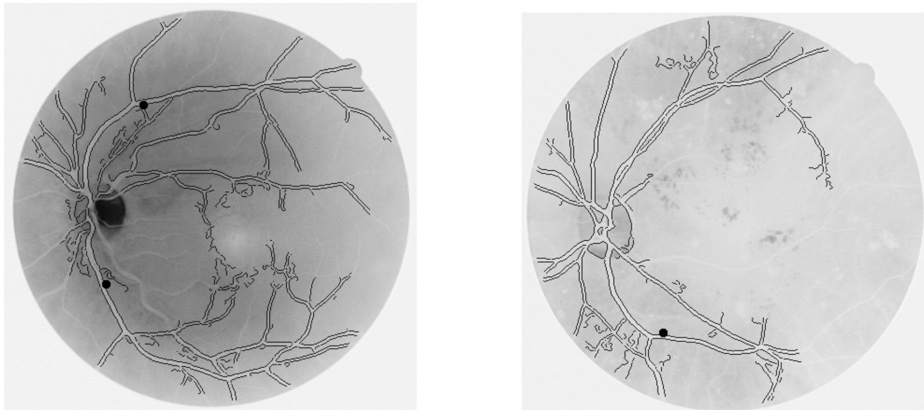

Figure 3: Tracking results for images D and E (no post-processing)

the edges being followed belong to a blood vessel. The next logical step is to develop additional algorithms for T-junction / bifurcation determination, since one major goal is to use branch positions as anchor points for image registration.

References

[1] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blod vessels in retinal images using two-dimensional matched filters. *IEEE Trans. Medical Imaging*, 8(3):263–269, 1989.

[2] G.W. Donohoe, P. Soliz, and S. Nemeth. Computer aided image analysis for background diabetic retinopathy. In *Proc. SPIE Medical Imaging*, 1998.

[3] M.A.T. Figueiredo and J.M.N. Leitao. A nonsmoothing approach to the estimation of vessel contours in angiograms. *IEEE Trans. Medical Imaging*, 14(1):162–172, 1995.

[4] M. Fontaine, L. Macaire, J-G. Postaire, M. Valette, and P. Labalette. Fundus image segmentation by unsupervised classification. In *Proc. Vision Interface'99*, pages 296–302, 1999.

[5] Z. Guo and R.W. Hall. Parallel thinning with two-subiteration algorithms. *Comm. of the ACM*, 32(3):359–373, March 1989.

[6] A. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. In *Proc. of the AMIA Annual Symposium 1998*, pages 931–935, 1998.

[7] B. Kochner, D. Schuhmann, M. Michaelis, G. Mann, and K.-H. Englmeier. Course tracking and contour extraction of retinal vessels from color fundus photographs: Most efficient use of steerable filters for model based image analysis. In *Proc. SPIE Medical Imaging 1998*, pages 755–761, 1998.

[8] L. Lecornu, J.-J. Jacq, and C. Roux. Simultaneous tracking of the two edges of linear structures. In *Proc. Int. Conf. on Image Processing*, pages 188–192, 1994.

[9] I. Liu and Y. Sun. Recursive tracking of vascular networks in angiograms based on the detection-deletion scheme. *IEEE Trans. Medical Imaging*, 12(2):334–341, 1993.

[10] F.P. Miles and A.L. Nuttall. Matched filter estimation of serial blood vessel diameters from video images. *IEEE Trans. Medical Imaging*, 12(2):147–152, 1993.

[11] V. Rakotomalala, L. Macaire, J-G. Postaire, and M. Valette. Identification of retinal vessels by color image analysis. *Machine Graphics and Vision*, 7:725–742, 1998.

[12] Y. Sun, R.J. Lucariello, and S.A. Chiaramida. Directional low-pass filtering for improved accuracy and reproducibility of stenosis quantification in coronary arteriograms. *IEEE Trans. Medical Imaging*, 14(2):242–248, 1995.

[13] Y.A. Tolias and S.M. Panas. An unsupervised fuzzy vessel tracking algorithm for retinal images. In *Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, volume 2, pages 325–330, 1997.

[14] P.W. Truitt, P. Soliz, D. Farnath, and S. Nemeth. Utility of color information for segmentation of digital retinal images: neural network-based approach. In *Proc. SPIE Medical Imaging*, 1998.

[15] F. Zana and J.-C. Klein. Robust segmentation of vessel from retinal angiography. In *Proc. Int. Conf. Digital Signal Processing*, pages 1087–1091, July 1997.

[16] F. Zana and J.-C. Klein. A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform. *IEEE Trans. Medical Imaging*, 18(5):419–428, 1999.

[17] L. Zhou, M.S. Rzeszotarski, L.J Singerman, and J.M. Chokreff. The detection and quantification of retinopathy using digital angiograms. *IEEE Trans. Medical Imaging*, 13(4), 1994.