Trace Inference, Curvature Consistency, and Curve Detection

Pierre Parent and Steven W. Zucker

IEEE Transactions on PAMI, Vol. 11, no 8, August 1989

Presented By:

Seniha Esen Yuksel

Department of Computer and Information Science and Engineering
University of Florida

May 7, 2007

Abstract

Upon our previous efforts to validate the DT-MRI fiber tracks with fibers extracted from fluoroscopy data, we fell into a problem of fiber crossings. Basic DT-MRI fiber tracking methods fail at the crossings and need to be inferred from the data. Hence, we found this paper useful for our research purposes, and implemented it.

This paper takes the inference procedure as a two step process; namely a trace inference stage and a curve synthesis stage; and focuses only on the first problem. Instead of taking simple first and second derivatives, the trace is estimated using constraints from tangents and curvatures, and from the number of constraints defined in a neighborhood depending on the tangents and curvatures.

Background

The trace of a curve is the image set in R^3 which is generated by the curve in a given interval. In other words, a curve is map (ie. a function, or equation), while a trace is just a picture of the curve. Hence, similar traces may be generated by different curves. For example, consider the two parabolas $a(t) = t, t^2$ and $b(t) = 2t, 4t^2$ shown in Fig. 1. Although both parabolas give the same trace, they are actually two different curves; as such, the velocity vector $a'(t)$ is half the length of $b'(t)$.

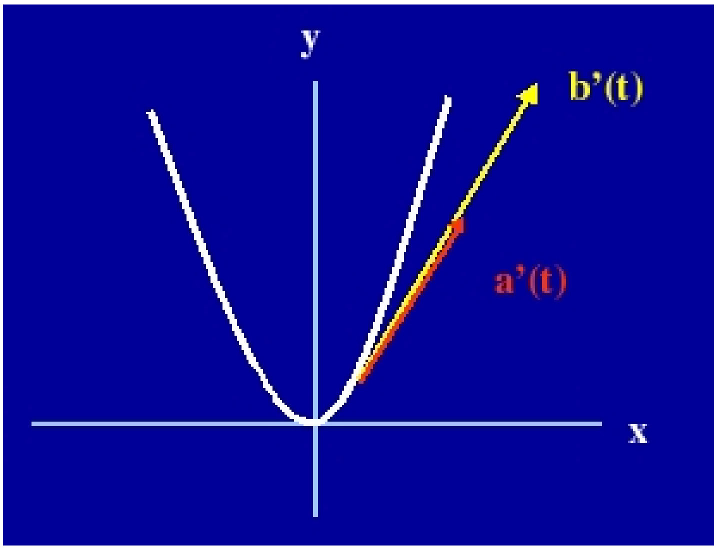


Figure 1: Two different parabolas with the same trace.

Hence, even if we have the trace points, we also need additional constraints to figure out the actual curve. Only then we can find the integral curves running through these fields subject to smoothness and discontinuity constraints. In Fig. 2, we see the same trace points give three different curves. In the first curve, the trace points are connected with positional constraints only; in Fig. 2(b), the trace points are connected with both positional and orientation (tangential) constraints; and in Fig. 2(c) the trace points are connected with both orientation, position and curvature constraints leading to different curves.

Therefore, if we know both the trace points and the position, orientation and curvature constraints; we can infer the true curve as shown in Fig. 3. Note that, as illustrated in the figure, at curve crossings or branches there are two or more tangential components; so it is important to know which tangential component belongs to which curve, yet it is more important to know if it is a significant tangential component or just noise. Hence, this paper is interested in finding these points and constraints by considering neighborhood relations; and it is left to another study to infer the actual curve.

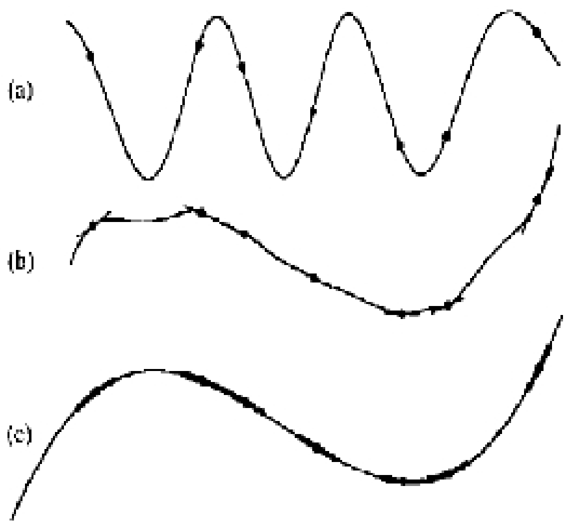


Figure 2: Given the trace points, you can fit a spline curve given the constraints: (a) positional constraints, (b) orientation and positional constraints, (c) orientation, position and curvature constraints.

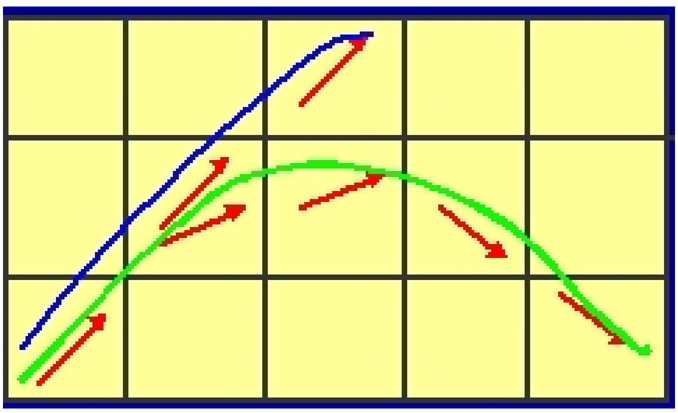


Figure 3: Given the trace points, orientation, position and curvature constraints; we can infer the true curve.

Trace Inference

In the rest of the report, we will be specifying constraints between (a) estimated tangents and (b) between curvature estimates.

As shown in Fig. 4; we will require that the neighboring points have consistent tangents and curvatures. With that consideration in mind, we will define a “support” parameter that will depend on tangential and curvature estimates. For a given point (x, y), if the neighboring points satisfy the constraints, they will be supporting the decisions for this point, hence the support parameter will increase; and it will decrease if otherwise. Hence, in the coming sections, we will first find the tangents, then find the curvatures and impose the constraints, and reach a better decision via relaxation labeling.

For the rest of the report, we will be showing our results on an angiogram image given in Fig. 5 which has several good features to investigate such as both thick and thin lines, crossings, branchings and noise.

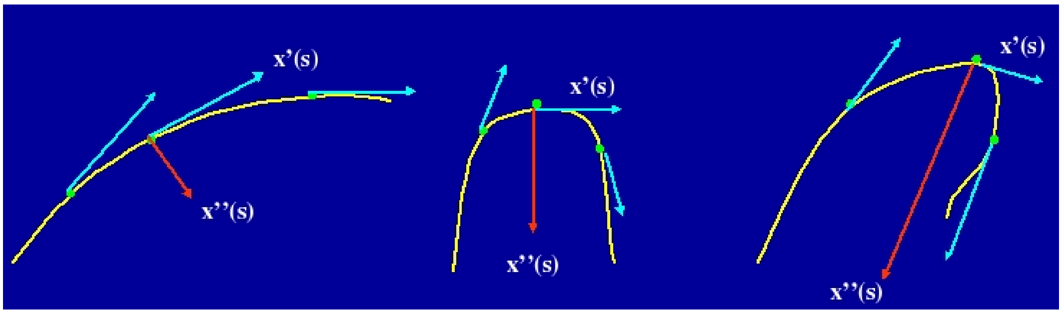


Figure 4: Given the trace points, orientation, position and curvature constraints; we can infer the true curve.

0.1 Tangent Estimates

Tangents are estimated by convolving the figure with a kernel given as:

$$G(x, y) = [Ae^{\frac{-x^2}{\sigma_1}} - Be^{\frac{-x^2}{\sigma_2}} + Ce^{\frac{-x^2}{\sigma_3}}] * e^{\frac{-y^2}{\sigma_y}} \tag{1}$$

The kernel is formed from one Gaussian in the tangent direction and the difference of three Gaussians in the perpendicular direction. With the parameters selected as: $A = 1$; $B = 1.266$; $C = 0.5$ $\sigma_1 = 2$; $\sigma_2 = 3.16$; $\sigma_3 = 4$; $\sigma_y = 5$; the kernels give a line detector.

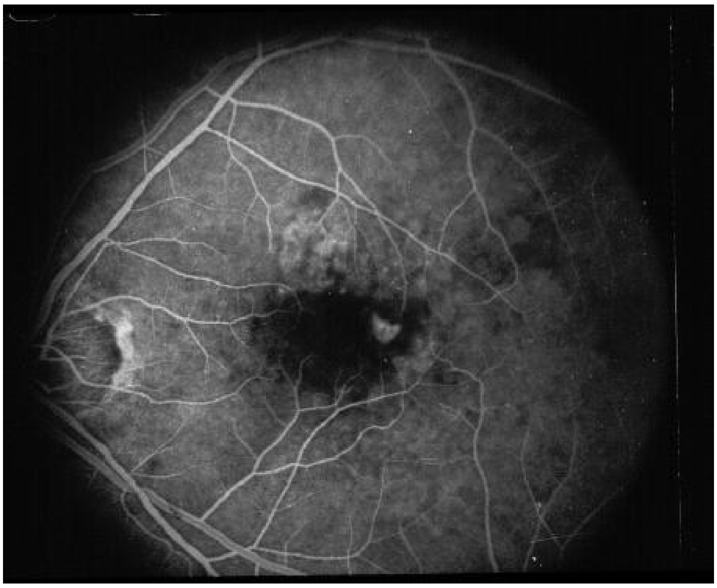


Figure 5: The original angiogram image.

With a simple transformation
$U = Xcos(\theta) + Ysin(\theta)$
$V = Ycos(\theta) - Xsin(\theta)$
in eight different orientations where θ ranges from 0 to $7\pi/8$; we get the rotated line detectors as shown in Fig. 6.

The convolution of these eight kernels with the image extracts the lines in the kernel direction as shown in Figs. 7 and 8. In Fig. 8, a simple image processing step is applied just for the visual purposes.

At the end of this stage, we have 8 tangent estimates at each pixel (x,y). When normalized, the tangent estimates give a certainty of having the tangent at each pixel at a particular orientation. In the paper, the tangent estimates are denoted as:
$p_i(\lambda)$ = certainty of tangent λ at position (x_i, y_i)
with
$\theta(\lambda)$ = discrete orientation of the tangent at a particular position $\lambda = 1...8$

If $p_i(\lambda) = 1$, the tangent λ is definitely associated with (x_i, y_i). As such, for curve points there is at least one element near 1, for noncurve points elements are near 0 and for curve crossings or orientation discontinuities, there are more than one near one values.

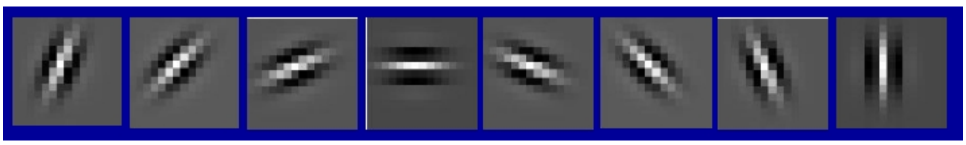


Figure 6: The eight kernels give line extractors.

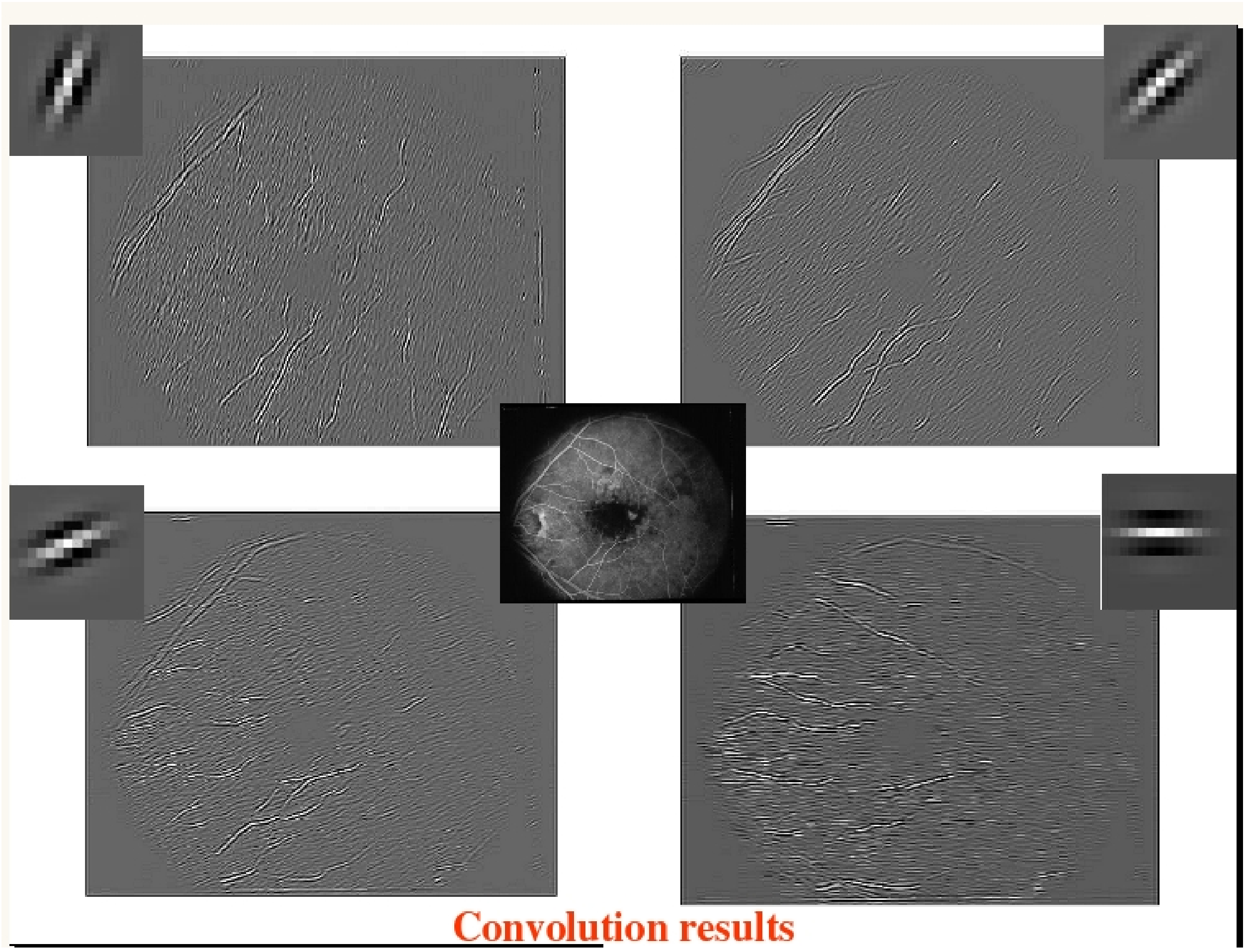


Figure 7: Convolution Results. The lines are extracted in the direction specified with the kernel. The image in the middle is the original.

0.2 Support

With the tangent estimates found in the previous section, we would like to *support* the estimates that are consistent with the neighbors' estimates. Hence, a "support function" is defined based on a set of constraints to help delete the noise and solidify the true tangents, and minimize curvature variation at each point by maximizing circularity. The support function is based on (i) cocircularity; (ii) curvature classes; (iii) curvature consistency and (iv) lateral maxima functions; where the cocircularity enforces the orientation constraints, curvature class constraint partitions the neighborhoods into regions and the curvature consistency term supports the estimates if the curvature class is similar to the neighbors' curvature classes. Finally, with the lateral maxima function, a positional constraint is introduced and line thinning is performed. All of these terms are combined into one single formula as shown in Fig. 9 and will be discussed in detail in the following subsections.

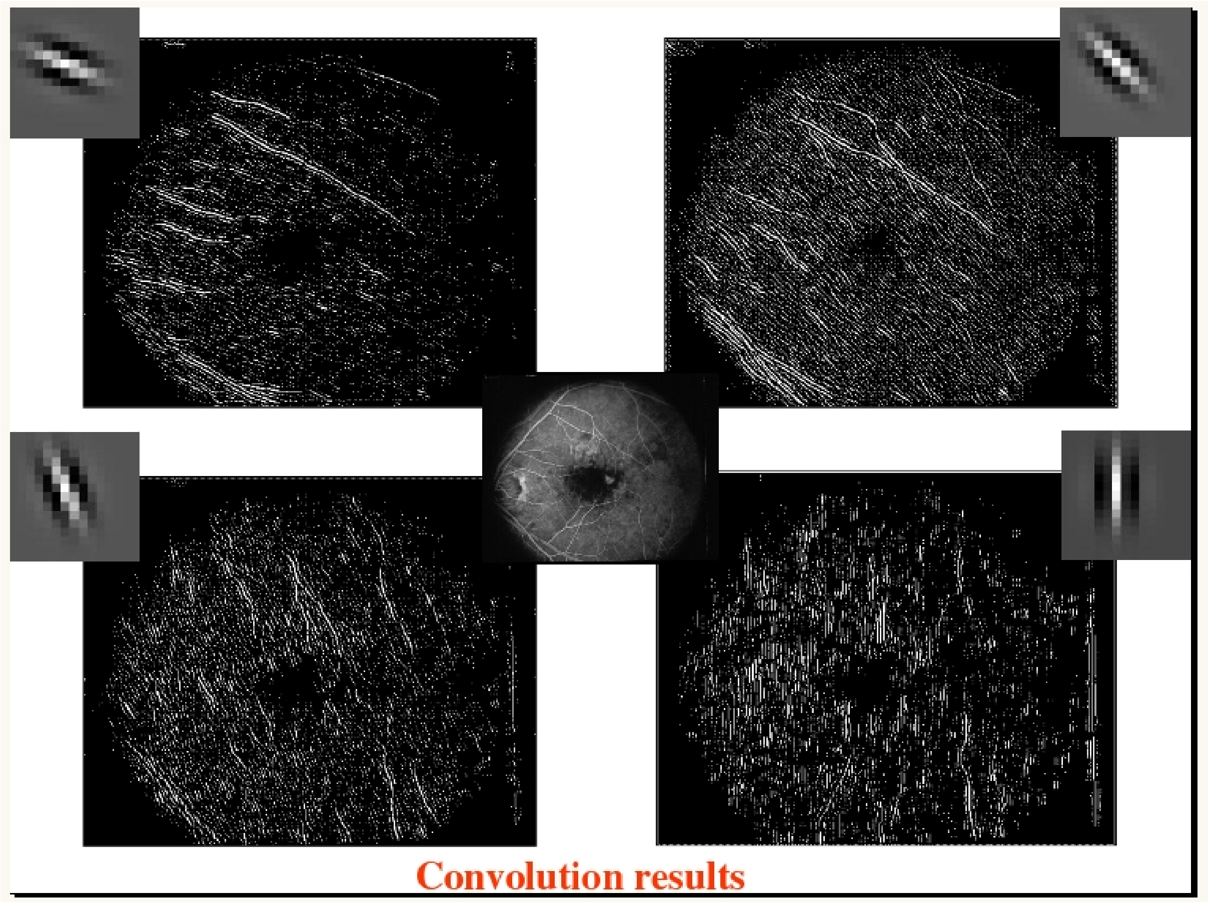


Figure 8: Convolution Results. The lines are extracted in the direction specified with the last four kernels. The image in the middle is the original. For better visualization, I did some image processing on these images.

0.2.1 Cocircularity

It is known that curvature estimates can be found by differentiating the curve twice; however, differentiation amplifies the noise. Hence, the paper chooses to find the curvature estimates with some geometrical constraints instead of differentiating twice; and introduces the term "cocircularity". Two point-tangent pairs are *cocircular* if the interior angle between the tangents and the line connecting the points are equal as shown in Fig. 10. In this figure, A and B are tangents to the same circle, and therefore the angles α and β are the same. Hence, this tangent pair is cocircular, denoted as $A \asymp B$

Let (x_i, y_i) and (x_j, y_j) be the coordinates of nodes i and j, let (x, y) be an arbitrary point within the circle of radius $1/2$ centered at (x_i, y_i), and let (x', y') be a point in a circular neighborhood of (x_j, y_j). Also, let λ and λ' be unit tangents at these locations with orientations θ_λ and $\theta_{\lambda'}$ as shown in Fig. 11. We wish to have the interior angles $\Gamma(\theta_\lambda, \theta_{ij})$ and $\Gamma(\theta_{ij}, \theta_{\lambda'})$ to be close to each other to satisfy the cocircularity constraint. If the constraint is relaxed to be continuous as shown in Fig. 12, we get the formula in the bottom of this figure. This constraint gives a 1 to the support

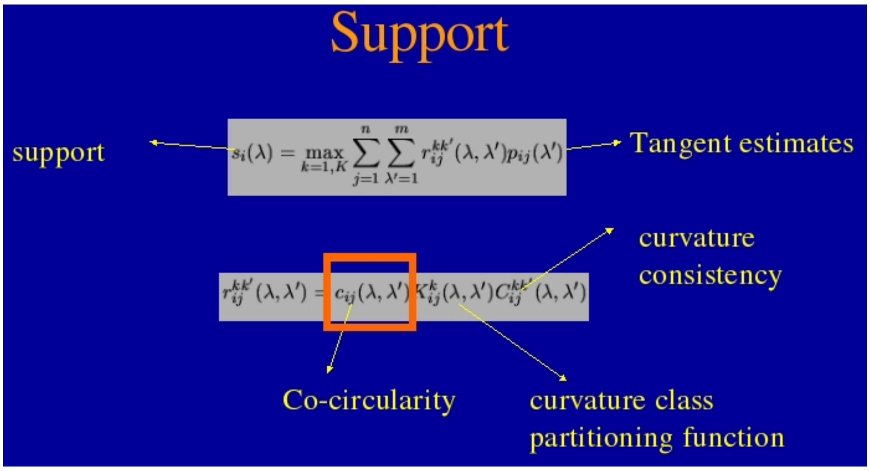


Figure 9: Support Formulation.

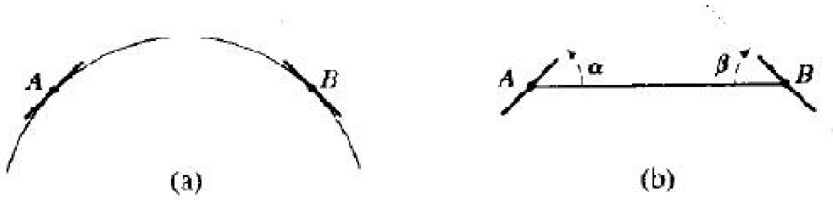


Figure 10: Cocircularity: A and B are tangents to the same circle, and therefore the angles α and β are the same. Hence, these two tangents are cocircular.

if these angles are sufficiently close, or smoothly assigns smaller values and provides less support.

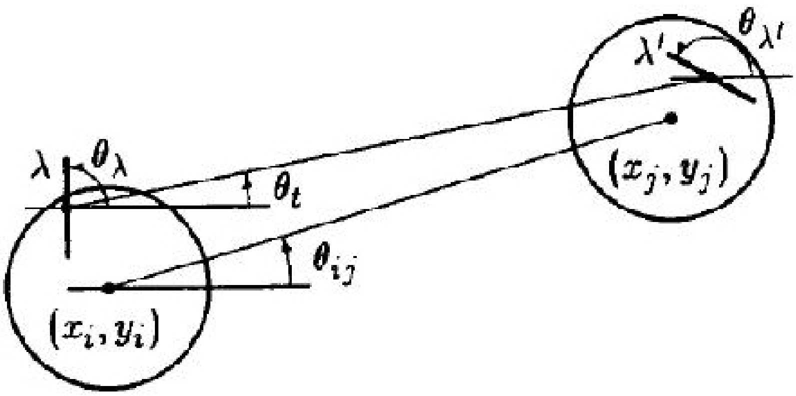


Figure 11: Cocircularity is defined on the closeness of the angles.

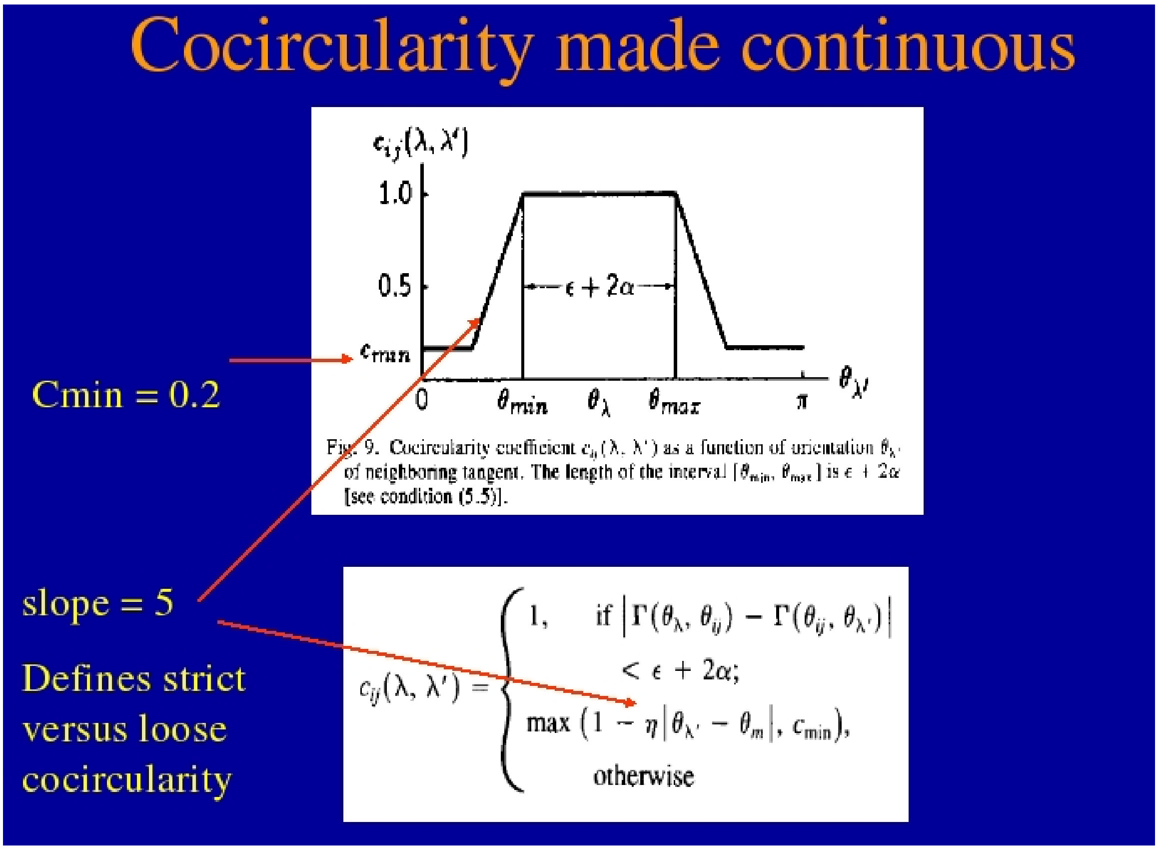


Figure 12: Cocircularity Formula

0.3 Curvature Class

In a small neighborhood of an image, there may be several tangent pairs that are mutually cocircular; however, $A \asymp B$ and $B \asymp C$ does not guarantee $A \asymp C$; hence, the neighborhood support set is partitoned into small curvature classes as shown in Fig. 13. The curvature is supported totally (i.e. multiplied by 1) if the curvature is between the specified curvature class limits; otherwise, the support is totally suppressed by multiplying with 0. This gives the second function in the support formula shown in Fig. 14.

0.4 Curvature Consistency

Even after partitioning into curvature classes, there might be some ambiguities as shown in Fig. 15. In Fig. 15(a), A and B are tangent to the same curve so they are cocircular. However in Fig. 15(b), although the spatial configuration is same, A and B are tangent to distinct curves and they are not cocircular. To eliminate such ambiguities, one point may support another if only if it falls within the extent of the other's curvature class, and this is formulated as in Fig. 16.

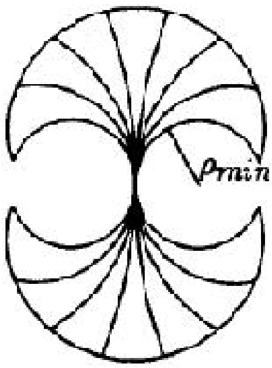


Figure 13: Curvature Classes

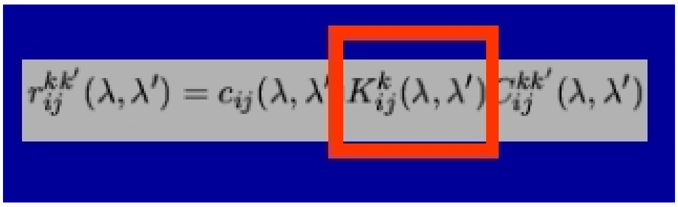


Figure 14: Curvature Class Formula

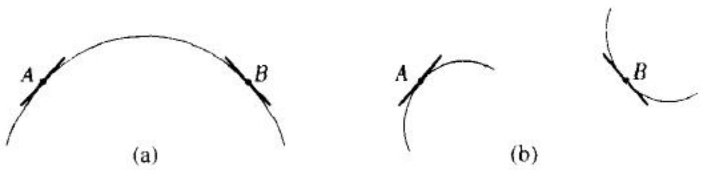


Figure 15: Curvature consistency is needed to eliminate ambiguities as in this figure. In (a), A and B are tangent to the same curves and hence cocircular. With the same spatial configuration in (b), A and B are not cocircular.

0.4.1 Lateral Maxima

The purpose of finding the lateral maxima is to prevent thickening of the curve by extracting a pixel-wide region about a curve as shown in Fig. 17. The support function gets its maximum value at the exact location of a curve and decreases gradually on either side of this location. Hence, the tangent λ at (x, y) is a lateral maxima if its certainty is maximal among all tangents in the 3 by 3 neighborhood of the point. With the lateral maxima as defined in Eq. 2; the maximal support points can be favored and others will be set to 0 to prevent thickening.

$$m_i(\lambda) = \begin{cases} 1 & \text{if } p_i(\lambda) > p_i(\lambda)'; \forall \lambda' \in \{\lambda - 1, \lambda, \lambda + 1\}, \forall j \in N_\lambda \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

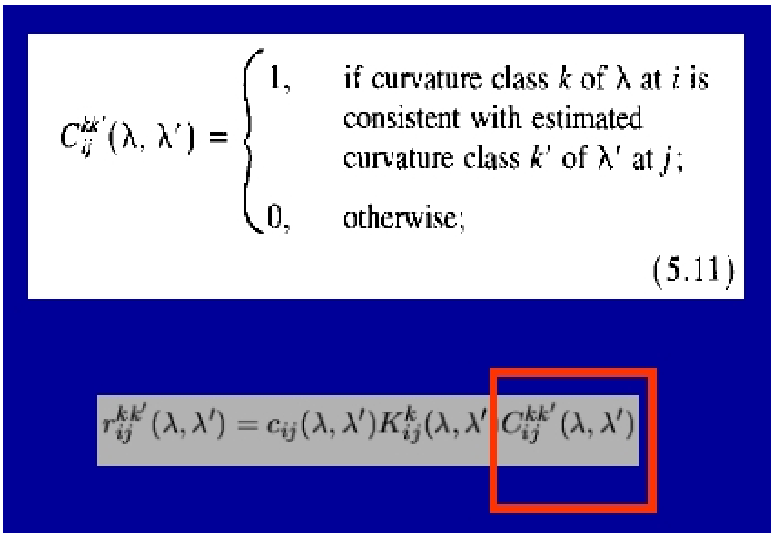


Figure 16: Curvature Consistency Formula

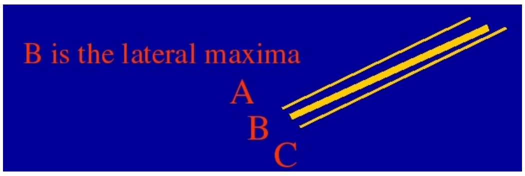


Figure 17: Given the A, B and C lines, B is the lateral maxima.

0.5 Relaxation Labeling

At the last step of the algorithm, relaxation labeling is used to maximize the average local support. With relaxation labeling, one looks for a consistent labeling of tangent estimates throughout the image through iterations. Relaxation labeling is performed in two steps; (i) calculate new probabilities, (ii) select the best probabilities. In these steps, a small step size of 0.01 is used and 2 iterations are performed.

The final support formula was given as:

$$s_i(\lambda) = max_{k=1,K} \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^{kk'}(\lambda, \lambda') p_j(\lambda') m_j(\lambda') \tag{3}$$

We would like to maximize this support with relaxation labeling iterations. The formula to calculate the new probabilities is given by:

$$sm_i^k = min_{\lambda=1,m} s_i^k(\lambda) \tag{4}$$

$$s_i^*(\lambda) = \begin{cases} s_i^k(\lambda) - sm_i^k & \text{if } sm_i^k(\lambda) \leq 0 \\ s_i^k(\lambda) & \text{otherwise} \end{cases} \tag{5}$$

Then the best probabilities are selected by:

$$p_i^{k+1} = \frac{p_i^k + s_i^{*k}}{1 + s_i^{*k}.1} \tag{6}$$

Results

In the following figures, we display the results on the upper right of the angiogram image because of time considerations. However, this part of the image contains useful amount of crossings and branchings with various contrast levels; therefore, I believe it is good enough in explaining the results.

The final tangent estimates are plotted with the quiver function of Matlab. Note that there are eight directions, and each color corresponds to one of the directions (except red – it is used twice because there are only seven colors in Matlab). Note that there are 8 tangent estimates at every pixel, with values between 0 and 1. These tangent estimates that have magnitude greater than 0.35 or 0.15 are plotted in the results shown below. Note that with the threshold selected as 0.35 we are able to get the strong tangent estimates in more defined regions; but with threshold set to 0.15 we get a chance to observe all the possible directions at the curve crossings and branchings.

I find it amazing to see the consistent tangent estimates especially at the crossings and branchings. Note that by the support constraints, we are able to get rid of all the noise that is appearant in Figs. 7 and 8 . I believe these results can really be very useful to infer the actual curves.

The algorithm takes very long time, the paper is not very easy to understand, there are too many parameters (around 17) to set and why the author combined all these formulas together becomes unintuitive at times; however, I think as a paper dated 1989, the results are really good.

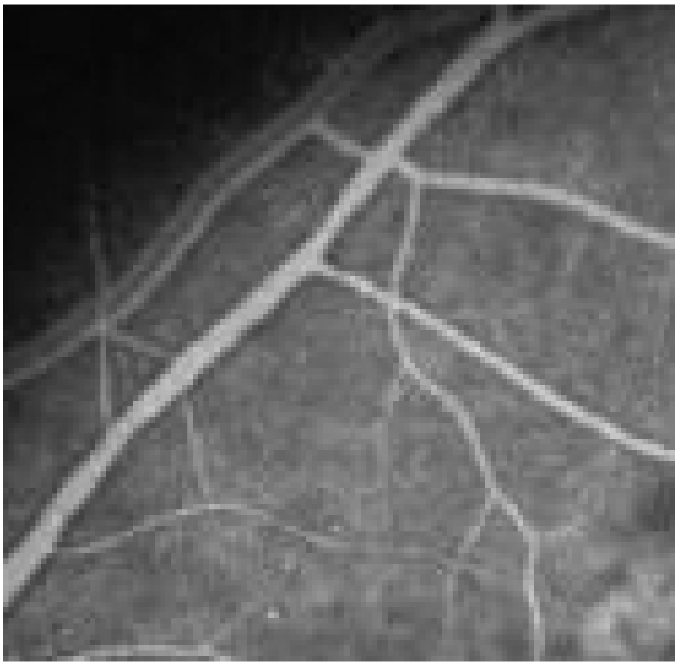


Figure 18: The upper right of the angiogram image is used for time considerations.

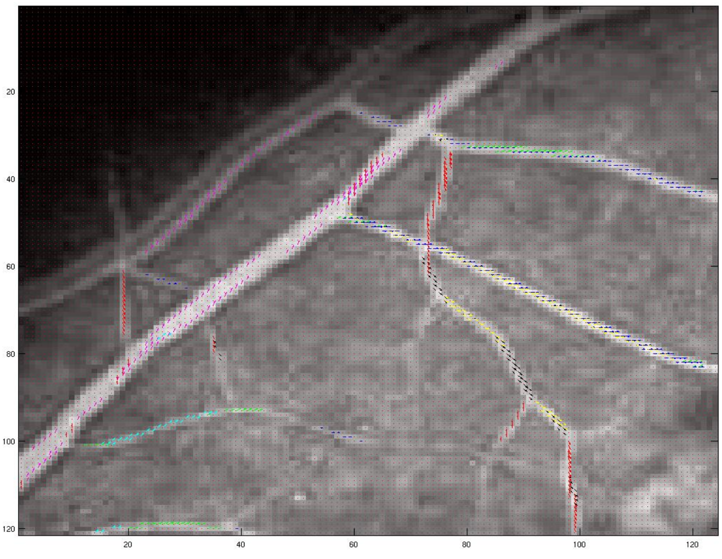


Figure 19: Tangent estimates in 8 directions, thresholded from 0.35. The tangent estimates give considerably good results and flow along the vessels.

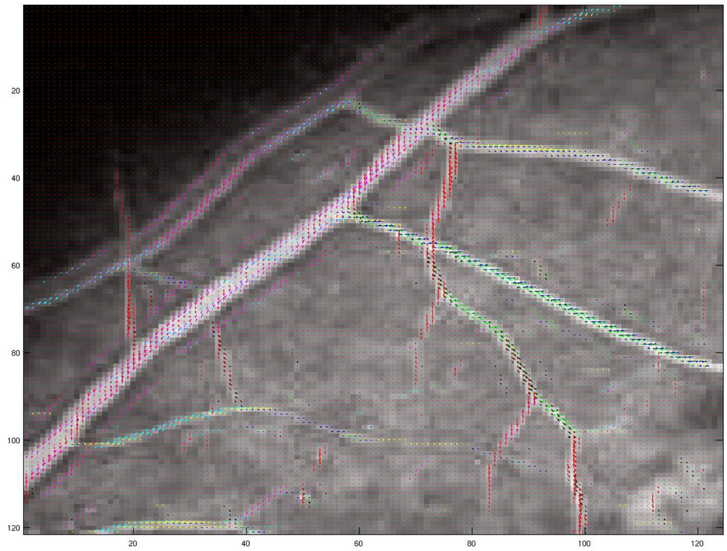


Figure 20: Tangent estimates in 8 directions, thresholded from 0.15. We observe more tangent estimates and this is especially helpful at the crossings and branchings.

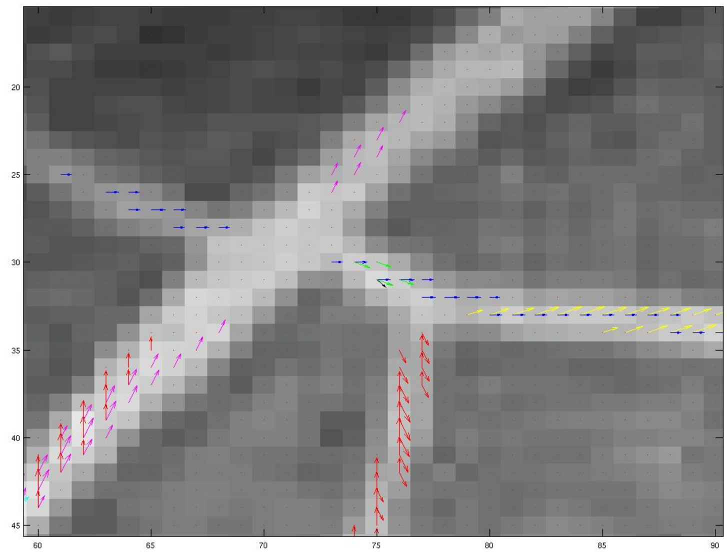


Figure 21: A vessel crossing with 0.35 threshold. Note that at the crossing there are both blue and green tangent estimates which further along the way become just the blue vector. At this crossing there are no appearant tangent estimates at the full crossing. This makes sense since there are tangent estimates at all directions, hence 0.35 is a big threshold.

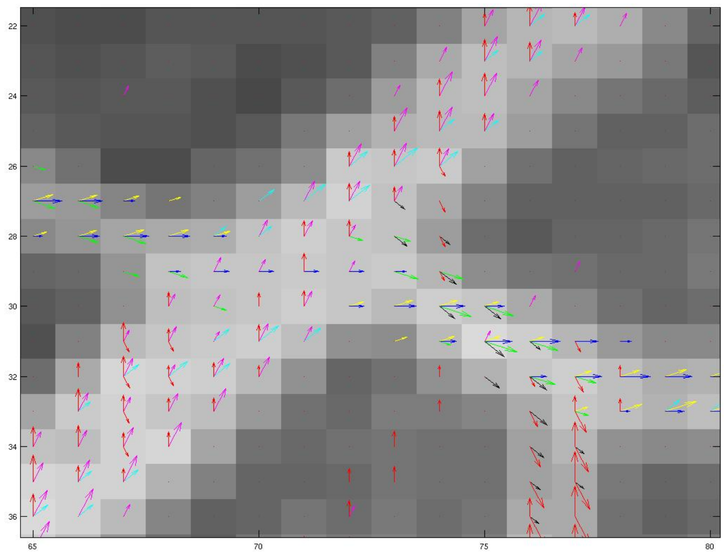


Figure 22: If the threshold is decreased to 0.15, we observe more of the tangents at the same crossing and get a good idea about the crossing.

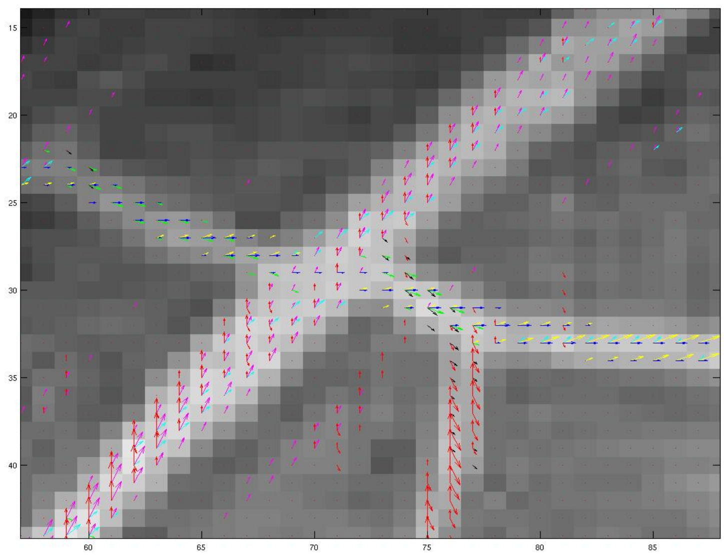


Figure 23: Same crossing with less zoom.

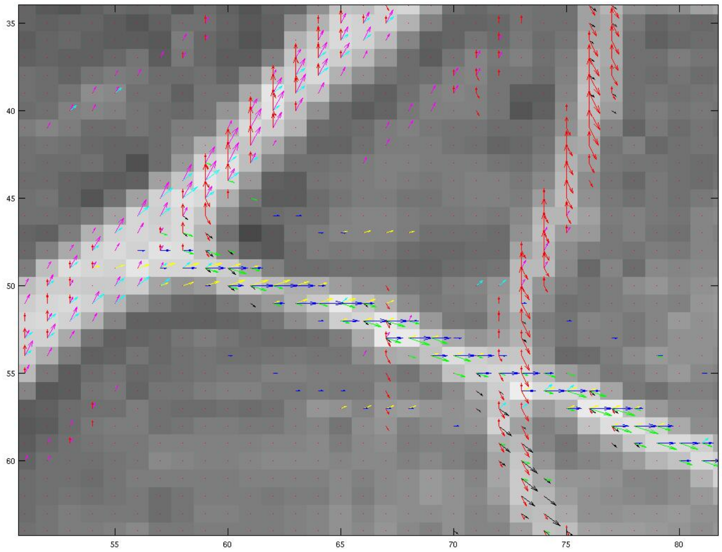


Figure 24: Another crossing with 0.15 threshold. Note the multiple tangent estimates at the intersections.

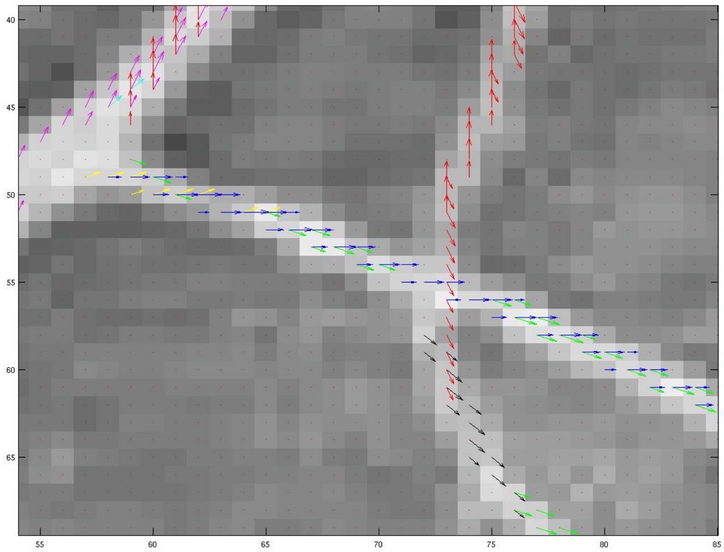
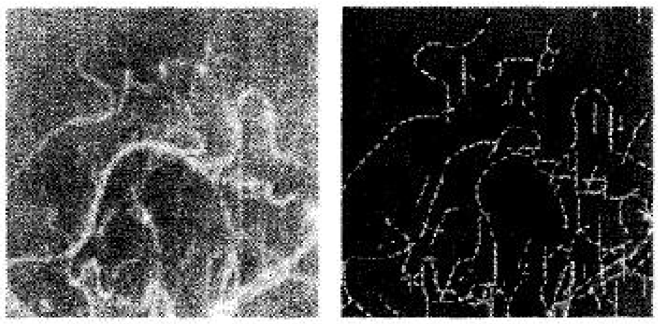
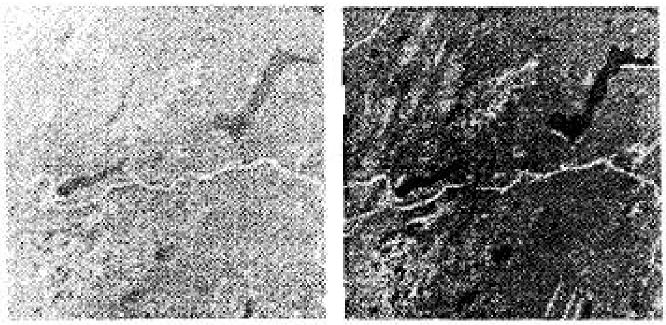


Figure 25: Yet another crossing with 0.35 threshold. The estimates are highly dependent on the threshold.



(a) (b)

Fig. 17. (a) An angiogram, or radiograph of blood vessels in the brain, and (b) the result of 2 iterations of the trace inference process over the image of an angiogram.



(a) (b)

Fig. 18. A satellite image of a forest with logging roads, and (b) the result of 2 iterations of the trace inference process, using a neighborhood size of 25, a step size of 1.0, and displayed at a confidence threshold of 0.6.

Figure 26: Two results from the paper itself.