# Interactive Retinal Vessel Extraction by Integrating Vessel Tracing and Graph Search

Lu Wang[1], Vinutha Kallem[1], Mayank Bansal[1], Jayan Eledath[1], Harpreet Sawhney[1], Karen Karp[2], Denise J. Pearson[3], Monte D. Mills[2], Graham E. Quinn[2], and Richard A. Stone[3,⋆]

[1] SRI International, Princeton, NJ, USA
[2] Children's Hospital of Philadelphia, Philadelphia, PA, USA
[3] Scheie Eye Institute, University of Pennsylvania, Philadelphia, PA, USA

**Abstract.** Despite recent advances, automatic blood vessel extraction from low quality retina images remains difficult. We propose an interactive approach that enables a user to efficiently obtain near perfect vessel segmentation with a few mouse clicks. Given two seed points, the approach seeks an optimal path between them by minimizing a cost function. In contrast to the Live-Vessel approach, the graph in our approach is based on the curve fragments generated with vessel tracing instead of individual pixels. This enables our approach to overcome the shortcut problem in extracting tortuous vessels and the problem of vessel interference in extracting neighboring vessels in minimal-cost path techniques, resulting in less user interaction for extracting thin and tortuous vessels from low contrast images. It also makes the approach much faster.

## 1 Introduction

Analysis of retinal images is useful for studying and diagnosing eye diseases, including retinopathy of prematurity (ROP) [1]. Accurate vessel extraction is a prerequisite for computing vessel tortuosity, width and branching patterns. Although many automatic approaches have been proposed, it remains a challenge to extract vessels from low contrast and noisy retinal images like those generally obtained from the eyes of premature infants. These retinal images may be captured using wide angle camera (RetCam) or one with a more limited field-of-view (Nidek NM-200D) [1]. With these cameras, due to optical and clinical factors [1], images are usually much lower in quality than the higher quality images obtained with cameras used for adults such as those images in the DRIVE dataset [2]. Figure 4(a) (b) are two typical examples. We are not aware of any existing automatic approach that can generate satisfying vessel extraction results on these images. This motivates us to seek a semi-automatic approach.

A traditional paradigm for vessel detection, which we refer to as vessel tracing in this paper, has 3 components. First is the computation of vesselness (likelihood of being a vessel) for each pixel with approaches such as matched filters [3],

⋆ Supported by NEI grant EY-017299, the Mackall Foundation Trust, and Research to Prevent Blindness.

K. Mori et al. (Eds.): MICCAI 2013, Part II, LNCS 8150, pp. 567–574, 2013.

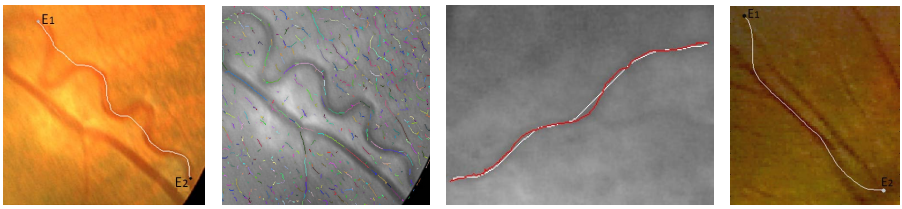
and Hessian-based filters [4]. Second is the vessel/non-vessel classification based on the vesselness measurement, e.g. by using a threshold. Last is the linking of the vessel pixels into curve segments based on connectivity. The problem with this paradigm is that the vessel/non-vessel classification is never perfect. There is always a trade off between false alarms and missed detection. Some approaches try to reduce false alarms by tracing vessels from seed points in combination with adaptive thresholding [3]. These seed points are manually selected in some interactive systems such as ROPTool [5]. However, vessel tracing is often halted prematurely or directed into wrong directions due to image noise. Therefore for low-quality images, the users of these interactive systems must provide a considerable number of seed points. Moreover, the extracted vessels are not globally optimal in terms of vesselness and smoothness.

Live-Vessel [4], developed by Poon, et al., is an efficient interactive vessel extraction approach based on global optimization. Given a starting point and the current cursor position, the approach finds the optimal path between the two points in a graph by minimizing a cost function with the Dijkstra's algorithm. In the graph, each pixel is a node and neighboring pixels are connected by edges. The cost function is a weighted sum of data terms measuring the vesselness of each pixel in the path and smoothness terms measuring the consistency of the vessel radius and orientation between neighboring pixels. The smoothness terms also encourage shorter paths in order to avoid contour jaggedness.

Although Live-Vessel has good performance on the DRIVE dataset in general, we found it has two important shortcomings. First is the shortcut problem due to the difficulty in reaching an optimum balance between the data terms favoring high vesselness and the smoothness terms encouraging shorter paths. This leads to difficulties in extracting highly tortuous vessels, as shown in Figure 1(a). The optimal path found by Live-Vessel between two endpoints $E_1$ and $E_2$ is far from the correct vessel. The shortcut problem also results in poor performance in extracting vessels that are not so tortuous but are of very low vesselness. Live-Vessel tends to straighten these vessels and make the extraction inaccurate. Figure 1(c) makes such an example. The vessel (white) output by Live-Vessel is too straight at the middle of the vessel compared to the ground truth (red).

Figure 1(d) illustrates another limitation of Live-Vessel in which a salient vessel can interfere with the extraction of nearby weaker vessels. The two seed points $E_1$ and $E_2$ do not yield the intended vessel. Instead, a path in which the two ends are composed of some non-vessel pixels and in which the middle part is related to a neighboring vessel with higher vesselness is returned as the minimal-cost path. Correcting these wrong or inaccurate vessel extractions caused by the two problems of Live-Vessel described above requires more user interactions.

In this paper, we propose an approach that can overcome the two problems of Live-Vessel without losing much of its advantages. The approach is based on the observation that the constraints of connectivity and linearity between vessel pixels that the traditional vessel tracing techniques rely on is not utilized in the global optimization scheme in which there is no vessel/non-vessel classification. Although this classification in vessel tracing is prone to noise and error, false




(a) (b) (c) (d)

**Fig. 1.** Limitation of Live-Vessel: (a) The minimum cost path between $E_1E_2$; (b) Curve segments from vessel tracing on (a); (c) The result of Live-Vessel (white) vs the ground truth (red); (d) A stronger vessel interferes with the extraction of a nearby vessel.

vessel pixels due to random noise usually do not align into a linear connected chain. Therefore, a long curve segment from vessel tracing has high possibility of being part of a vessel, even if it is very tortuous and each of its pixels has very low vesselness. This can be seen in Figure 1(b) in which the main structure of the tortuous vessel is detected by vessel tracing. Although it is broken into many curve fragments, the fragments line up into a linear chain. On the other hand, the global optimization scheme is powerful in filling the gaps between the fragments and grouping them into a continuous vessel. Therefore, we integrate these two complimentary techniques. First vessel tracing is applied to extract curve fragments. Then, to link the fragments into a vessel path between two seed points, a minimal-cost path is searched for in a graph whose nodes represent the individual curve fragments and the edges represent their connections.

The key in the vessel tracing is to keep the missed detection rate low whereas false alarms can be pruned in the graph optimization. The cost function in the graph optimization is designed to maximize the connectivity and linearity of the path, by minimizing the gap distance between the curve fragments and by maximizing the smoothness of the connection between the fragments respectively. There is no extra penalty on the total length of the path nor on the tortuosity of the curve fragments so that the approach can overcome the shortcut problem. The cost function also does not depend on the vesselness measurement, so salient vessels will have less interference on extraction of nearby weak vessels. In addition, the optimization in this approach is much faster than that in Live-Vessel due to far fewer nodes in the graph.

**Related Work:** The shortcut problem of minimal path techniques has also been pointed out by other researchers [6]. Zhu [6] proposed using minimal average-cost path instead of minimal-cost path. However, the approach can aggravate the problem of interference between neighboring vessels since long and strong vessels become more dominant in reducing the average cost. Wang [7] proposed an interactive approach for guidewire extraction in 2D Fluoroscopy by finding a minimal cost path in a graph structure constructed based on curve segments. However, their curve segments are short line segments with constant length. In our approach, the curve segments are usually long and with arbitrary shape so that the connectivity constraint in vessel tracing is fully utilized. More

importantly, solving the shortcut problem is not targeted in [7] and their cost function gives significant penalties to long and tortuous vessels, which leads to similar difficulties as those of Live-Vessel. Many existing perceptual grouping works also studied the problem of linking short edges into longer edges. However, most of them aimed at automatic object boundary extraction and many assumed contour closure[9]. Our grouping approach is guided by user interaction and is optimized for extracting vessels that are usually open curves.

The remainder of this paper is organized as follows: Section 2 describes the preprocessing and vessel tracing. Section 3 gives details about the vessel extraction by finding the optimal path. Section 4 presents the experimental results on DRIVE and two of our ROP datasets, and the paper is concluded in Section 5.

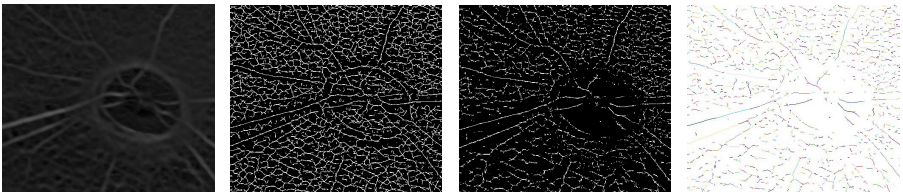
## 2 Preprocessing and Vessel Tracing

Our system adopts the approach of Sofka, et al [3], for vesselness computation. Other vesselness computation approaches can also be applied. The vesselness of a pixel is its maximum response of a matched filter over multiple scales and orientations. Figure 2(a) is an example of a vesselness map. The orientation and radius of the matched filter at the maximum response are the vessel orientation and radius at the pixel. Many non-vessel pixels can be removed with non-maximum suppression on the vesselness map as shown in Figure 2(b).

The matched filter has strong response not only to vessels but also to step edges [3]. To reduce false alarms, we check the edge response on both sides of the vessel boundary. In particular, let $\mathbf{x}$ denote the pixel position, $\mathbf{n}$ the normal to the vessel orientation and $W$ the vessel width. At each of the two potential edge locations $\mathbf{x_1} = \mathbf{x} + (W/2)\mathbf{n}$ and $\mathbf{x_2} = \mathbf{x} - (W/2)\mathbf{n}$, we search in a $\pm 1$ pixel interval along $\mathbf{n}$ to find if there is a pixel whose gradient orientation, when projected onto $\mathbf{n}$, points outwards from $\mathbf{x}$. We only keep the pixels that pass this test and have positive matched filter response (blood vessels are dark). Figure 2(c) shows the result after this verification. Unlike some other approaches, we do not set a threshold on the vesselness of a pixel nor the gradient magnitude of its supporting edge pixels in order to lower the missed detection rate. The vessel pixels are then linked into curve segments with edge tracing based on the 8-connected neighborhood. The segments shorter than 3 pixels are removed to reduce noise. Figure 2(d) shows the curve segments after vessel tracing.

## 3 Vessel Extraction by Finding an Optimal Path

Given the two endpoints of a vessel input by a user, our approach extracts the vessel by finding the optimal path between them. The path is formed by connecting the curve segments generated with vessel tracing. An example is shown in Figure 3(a), where $E_1$ and $E_2$ are the two endpoints. A path with higher connectivity and linearity, meaning shorter gaps and smoother connections



(a) (b) (c) (d)

**Fig. 2.** Pre-processing and vessel tracing, applied to the image in Fig. 4b: (a) Maximal response of the matched filter; (b) Non-maximum suppression; (c) Vessel verification; (d) Curve fragments from vessel tracing shown in random color

between the curve fragments, is more likely to be the correct vessel. Therefore, our approach seeks a path that can minimize the following cost function:

$$c = \sum_{i=1}^{N-1} g_i + w g_i((\alpha_i/\pi)^p + (\beta_i/\pi)^p), \tag{1}$$

where $N$ is the number of curve segments in the path. The total cost is the sum of the connection cost of every segment with its following segment in the path; the cost has two terms in Equation 1: the length of the gap $g_i$ and the second term representing the connection smoothness gauged by the continuity of tangent orientation. As shown in Figure 3(b), assuming that $\overrightarrow{AB}$ is the $i$-th segment and $\overrightarrow{CD}$ is its following segment (overhead arrow indicates the direction of the curve segment in the path), $g_i$ is the length of the gap $BC$, and $\alpha$ and $\beta$ are the angles from the orientation of line $\overrightarrow{BC}$ to the tangent orientation of segment $\overrightarrow{AB}$ at $B$ and that of $\overrightarrow{CD}$ at $C$ respectively. Note that the smoothness term is proportional to the gap length so the tolerance to the orientation discontinuity depends on the gap length. In Equation 1, $w$ and $p$ are parameters. In our experiments $w = 6$ and $p = 2$, which are learnt by maximizing the performance on a set of training data with manual tracing as the ground truth.

The two endpoints of the path are also treated as two curve segments with 0 length. One is selected as the starting point for which we set $\alpha = 0$ in Equation 1; and for the other point, $\beta = 0$. To find the optimal path, a directed graph is created. Each curve segment generates two nodes in the graph corresponding to its two possible directions in the path. Every two nodes of different segments are connected with two edges representing the connection between the two segments in two possible orders, and the edge cost equals the connection cost. The path with the minimal cost can be found efficiently with Dijkstra's algorithm.

In practice, however, the linking of vessel pixels into curve segments by vessel tracing contains errors, especially around endpoints and junctions. This sometimes makes connecting curve segments by directly linking their endpoints as in Figure 3(b) unreasonable. As illustrated in Figure 3(c), the connection cost of $\overrightarrow{AB}$ with $\overrightarrow{CD}$ is very high if we link the endpoint $B$ with $C$ because the connection is not smooth. However, the cost can be much smaller if the linking is from

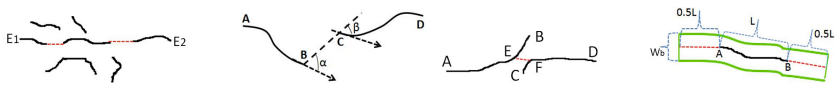
$E$ to $F$ that are internal points on the segments. Considering the possible error in vessel tracing, we choose the path $AE \rightarrow FD$ instead of $AB \rightarrow CD$.

Therefore, before the graph is created, we find the optimal connection from segment $\overrightarrow{AB}$ to $\overrightarrow{CD}$ using the following approach. First, we find the points $E'$ and $F'$ that have the smallest distance between the two curve segments. Then, we search in a $\pm 5$ pixels interval centered at $E'$ and $F'$ on the two curve segments for the points $E$ and $F$ so that the connection cost of $AE \rightarrow FD$ based on Equation 1 is minimal. The reason to limit the search range is for efficiency. To further reduce computation, we calculate the optimal connection only for neighboring curve segments. The range of the neighborhood of a curve segment is proportional to its length as depicted in Figure 3(d).

If the optimal connection requires breaking one of the segments, new curve segments will be formed. For Figure 3(c), new segments $AE$ and $FD$ will be added to the graph. However, $\overrightarrow{AE}$ can only be followed by $\overrightarrow{FD}$ in the path, and $\overrightarrow{DF}$ can only be followed by $\overrightarrow{EA}$, since their existence depends on each other. In addition, the connection from $\overrightarrow{AB}$ to $\overrightarrow{CD}$ will be removed from the graph.

Once the optimal path is found, to bridge the gap between two consecutive curve segments, we can use the same approach as Live-Vessel to find the optimal path between the two endpoints of the gap on a graph with each pixel as a node. However, in our experiments, this has no significant improvement over simply filling the gap with a straight line segment. We think this is because, in the areas where vessel tracing fails to find the curve fragments, the smoothness term encouraging the shortest path in the cost function of Live-Vessel becomes dominant over vesselness measurement. For the sake of efficiency, we simply link the gaps with straight line segments.

The vessel extracted based on only two endpoints is not always ideal. A user can make corrections by providing a few more seed points on the vessel until the result is optimal. The final vessel is the concatenation of the optimal path between every two spatially consecutive seed points on the vessel. Once the centerline of a vessel is obtained, each pixel on the centerline gives two pixels on the boundary of the vessel with $\mathbf{x_b} = \mathbf{x_c} \pm (W/2)\mathbf{n}$, where $\mathbf{x_c}$ is the position of the center pixel, $W$ is its vessel width and $\mathbf{n}$ is its vessel normal. These boundary pixels define the contour of the vessel boundary on each side of the vessel but they can be noisy. Therefore, we use a similar approach to [8] to fit a smooth curve to these boundary pixels with RBF kernel regression.




(a) (b) (c) (d)

**Fig. 3.** Vessel extraction: (a) The path between $E_1$ and $E_2$; (b) The cost computation for connecting AB to CD; (c)The optimal connection from AB to CD ; (d) The neighborhood of AB is defined by the green contour ($W_b$= 40 pixels)

## 4 Experiments

We tested the approach on the DRIVE dataset [2] of retinal images from adult eyes, and retinal images from premature infants at risk for ROP using 15 RetCam images and 15 Nidek images. Figure 4(a) and (b) are examples from our RetCam and Nidek ROP datasets. In Figure 4(c), the extraction is perfect based on the two endpoints given by a user even though the vessel is very tortuous. Figure 4(d) shows another example given only the two endpoints of a vessel. Notice that the extraction of the part near the lower endpoint is correct without interference from the nearby much stronger vessel. In Figure 4(e), the user obtained the target vessel by adding another seed point.

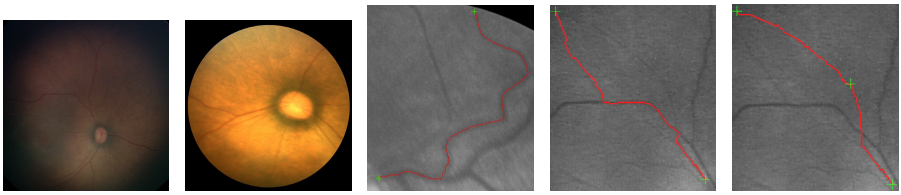
The performance of the approach is quantified with three criteria: reproducibility, accuracy, and efficiency, which also are the criteria used in [4]. Reproducibility is measured by comparing the results of several trials of the interactive segmentation on the same images. For each image, 3 trials are performed. The reproducibility rate is calculated as the average pairwise Dice similarity between different trials with the equation: $s = \frac{2|X \cap Y|}{|X|+|Y|}$, where $X$ and $Y$ are the sets of vessel pixels on the two segmentations in comparison. The rates are 0.993, 0.990, 0.991 for the DRIVE, RetCam and Nidek datasets respectively, which are high and similar to those of Live-Vessel, meaning that the difference in the seed point selection does not greatly impact the vessel extraction results.

Following [4], manual vessel segmentation is used as the ground truth to quantify accuracy. For each dataset, we have manual tracings from two expert graders. Manual tracing on the same image can be different between different human operators due to two factors: 1) they segment the same vessels differently, mainly on the vessel boundaries that can be ambiguous; 2) they select different vessels to trace. To compute the similarity between two vessel segmentations, we eliminate the second factor by manually removing the vessel branches that are not in both of the segmentation results. Table 1 gives the average Dice similarity between the two manual tracings, and between the interactive segmentation and every manual tracing for each dataset. We can see that both the accuracy of our approach and that of Live-Vessel are comparable to a human operator.

Efficiency can be measured as the time required for a user to complete the interactive vessel extraction. It excludes the time spent in preprocessing, such as computing filter response, since that can be done automatically offline. Table 1 shows that compared to Live-Vessel our approach significantly reduced the time by 62%, 65% and 77% for the DRIVE, RetCam and Nidek datasets respectively.

## 5 Conclusion

In this paper, we present an interactive approach for vessel extraction by integrating local vessel tracing and global graph optimization. Our approach was validated with three datasets. It demonstrated high accuracy, high reproducibility, and significant reduction on user interaction compared to Live-Vessel. In particular, the approach improves the efficiency in extracting tortuous and low contrast vessels that are common in retinal images from premature infants.



(a) (b) (c) (d) (e)

**Fig. 4.** (a) A wide angle RetCam image; (b) A narrow angle Nidek image; (c) The extraction result given only the two endpoints; (d) The result given only the two endpoints; (e) The result of (d) is corrected by adding another seed point on the vessel.

**Table 1.** T1 and T2 represent manual tracing 1 and 2. LV represents Live-Vessel. The efficiency is measured based on the average interaction time spent on each image.

| | Accuracy (Dice similarity) | | | | | Efficiency | | |
|---|---|---|---|---|---|---|---|---|
| | T1-T2 | LV-T1 | LV-T2 | Ours-T1 | Ours-T2 | LV | Ours | Time reducion |
| DRIVE | 0.814 | 0.820 | 0.811 | 0.823 | 0.815 | 11m:14s | 4m:16s | 62% |
| RetCam | 0.799 | 0.790 | 0.801 | 0.806 | 0.797 | 3m:11s | 1m:06s | 65% |
| Nidek | 0.808 | 0.809 | 0.798 | 0.812 | 0.806 | 4m:30s | 1m:00s | 77% |

## References


1. Wilson, C.M.: Classification and Automated Detection of the Retinal Vessels of Premature Infants with and without Retinopathy of Prematurity. PhD Thesis (2009)
2. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 23, 501–509 (2004)
3. Sofka, M., Stewart, C.: Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. IEEE Transactions on Medical Imaging 25, 1531–1546 (2006)
4. Poon, K., Hamarneh, G., Abugharbieh, R.: Live-vessel: Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 444–451. Springer, Heidelberg (2007)
5. Wallace, D.K.: Computer-assisted quantification of vascular tortuosity in retinopathy of prematurity. Transactions of the American Ophthalmological Society 105, 594–615 (2007)
6. Zhu, N., Chung, A.C.S.: Minimum average-cost path for real time 3D coronary artery segmentation of CT images. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011, Part III. LNCS, vol. 6893, pp. 436–444. Springer, Heidelberg (2011)
7. Wang, P., Liao, W.-S., Chen, T., Zhou, S.K., Comaniciu, D.: Graph based interactive detection of curve structures in 2D fluoroscopy. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part III. LNCS, vol. 6363, pp. 269–277. Springer, Heidelberg (2010)
8. Bansal, M., Kuthirummal, S., Eledath, J., Sawhney, H., Stone, R.: Automatic blood vessel localization in small field of view eye images. In: International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5644–5648 (2010)
9. Stahl, J.S., Wang, S.: Edge Grouping Combining Boundary and Region Information. IEEE Transactions on Image Processing 16, 2590–2606 (2007)