Direction-Based Segmentation of Retinal Blood Vessels

M. Frucci[1], D. Riccio[1,2], G. Sanniti di Baja[1], and L. Serino[1(✉)]

[1] Institute for High Performance Computing and Networking, CNR, Naples, Italy
{maria.frucci, gabriella.sannitidibaja, luca.serino}@cnr.it, daniel.riccio@unina.it
[2] University of Naples Federico II, Naples, Italy

Abstract. An unsupervised method is introduced for retinal blood vessels segmentation. The direction map is built by assigning to each pixel a discrete direction out of twelve possible ones. Under- and over-segmented images are obtained by applying two different threshold values to the direction map. Almost all foreground pixels in the under-segmented image can be taken as vessel pixels. Missing vessel pixels in the under-segmented image are recovered by using the over-segmented image. The method has been tested on the DRIVE dataset [1] producing satisfactory results, and its performance has been compared to that of other unsupervised methods.

Keywords: Retinal image · Blood vessel segmentation · Direction map

1 Introduction

Automatic procedures to analyze retinal blood vessels are useful in ophthalmology to allow an early diagnosis of a number of diseases, such as diabetic retinopathy, arteriosclerosis, hypertension, cardiovascular diseases and stroke [2, 3]. The automatic analysis of the structure of retinal vessels is also useful in biometrics [4], since the structure is different for each individual and even for the left and the right eye of the same person. In the literature, segmentation techniques based on matched filters, e.g., [5], wavelet transform, e.g., [6, 7], line detectors, e.g., [8–12], and morphological image processing [8, 9, 13–16] are available. In this paper we present an unsupervised retinal blood vessels segmentation method based on directional information. Each pixel of the green channel G of a RGB retinal image is assigned a direction out of twelve possible discrete directions. Two different threshold values are then employed to roughly segment the so obtained direction map DM_G. The foreground in the under-segmented image G_u, obtained in correspondence with the higher threshold value, includes pixels that can be most possibly interpreted as vessel pixels. The foreground in the over-segmented image G_o, obtained in correspondence with the lower threshold value, includes several more pixels than G_u. Some pixels of G_o are detected as missing vessel pixels of G_u, and are added to G_u to improve the quality of the segmentation. The method has been tested on the DRIVE database [1] producing satisfactory results, and its performance has been compared to that of other unsupervised methods.

C. Beltrán-Castañón et al. (Eds.): CIARP 2016, LNCS 10125, pp. 1–9, 2017.
DOI: 10.1007/978-3-319-52277-7_1

2 Building the Direction Map DM_G

We work with the green channel G of RGB color retinal images, as done by the majority of authors in the field, since G is the channel characterized by the highest contrast. Gray-levels in G are in the range [0,255], where lighter pixels are those with larger gray-level values. Vessel pixels are thin structures whose pixels are generally darker than their neighboring non-vessel pixels and are aligned along different directions.

The direction of any pixel p of G can be computed by taking into account the gray-levels of the pixels in an $n \times n$ window W centered on p, so as to build the Direction Map DM_G. Selecting the proper value for n is crucial to obtain a correct segmentation: W should be large enough to include both vessel and non-vessel pixels, even when centered on the most internal pixels of the thickest vessels, so as to have appreciable variations of gray-levels within the window; on the other hand, W should not be too large so as to avoid the inclusion of vessel pixels belonging to close vessels. By taking into account the average width and distance of vessels in the DRIVE database, we set $n = 7$. Since in an $n \times n$ window, $2 \times (n-1)$ discrete straight lines can be built, 12 directions with an angle of 15° between any two successive directions are obtained. See Fig. 1 top, where the twelve directions are shown in different colors.

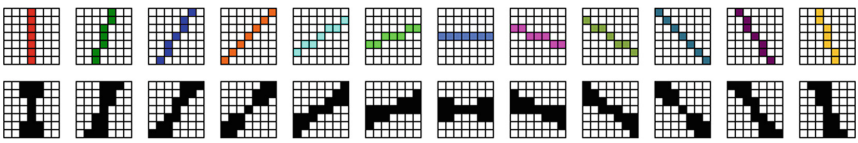

Fig. 1. The twelve directions d_i, top, and the twelve directional templates T_d_i, bottom. (Color figure online)

To build DM_G we use the twelve 7×7 directional templates T_d_i, $i = 1,2,\ldots,12$, shown in Fig. 1 bottom. In the i-th template, the pixels aligned along any direction out of d_{i-1}, d_i, and d_{i+1} are set to 1, and all the remaining pixels are set to 0. Of course, for $i = 1$ ($i = 12$) d_{i-1}, d_i, d_{i+1} are respectively d_{12}, d_1, d_2 (d_{11}, d_{12}, d_1).

Given two arrays $DIFF$ and DM_G with the same size as G and initially empty, for each pixel p of G and for any directional template T_d_i, $i = 1,2,\ldots,12$, we compute:

- the arithmetic mean, $Ad_i(p)$, of gray-levels of pixels of G matching 1's in T_d_i,
- the arithmetic mean, $NAd_i(p)$, of gray-levels of pixels of G matching 0's in T_d_i,
- the difference $\Delta_i(p) = NAd_i(p) - Ad_i(p)$.

The direction d_i for which $\Delta_i(p)$ has the maximum value $M(p)$ is taken as direction of p; the score $M(p)$ and the index i are respectively stored in the homologous position of p in $DIFF$ and DM_G. For the green channel G of the image 05_test of the DRIVE database, shown in Fig. 2 left, the obtained DM_G is shown in Fig. 2 middle, where colors correspond to the directions as in Fig. 1 top.

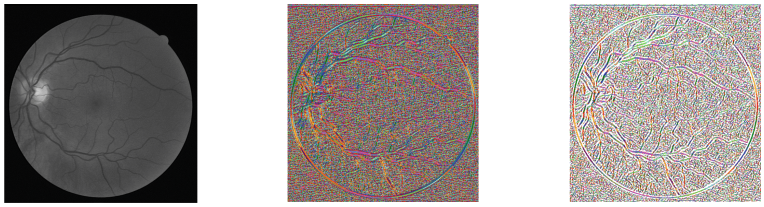

Fig. 2. The image G, left; the direction map DM_G before, middle, and after, right, direction updating. (Color figure online)

Actually, some pixels in DM_G have been assigned a direction different from the one along which they appear to be aligned. Thus, an updating of DM_G is done for each of the twelve directions and by using an auxiliary array AUX with the same size as G and whose pixels are initially set to 255. For the i-th direction d_i, the pixels of DM_G with assigned direction d_i are set to 0 in AUX. Thus, AUX becomes a gray-level image with only two values, where the foreground consists of the pixels with direction d_i, and the background includes all the remaining pixels. The idea is to build the direction map of AUX, DM_{AUX}, by following the procedure described to compute DM_G and to compare the direction assigned to any pixel p in the two maps. Only pixels for which the same direction is obtained in DM_G and DM_{AUX} are confirmed as foreground pixels in DM_G. All other pixels of DM_G are assigned to the background. Actually, when transferring in AUX pixels with direction d_i, also the pixels that in DM_G where characterized by the directions d_{i-1} and d_{i+1} are set to 0 in AUX, while updating is done only for the pixels with direction d_i. The purpose is to avoid that only a few sparse pixels of AUX are foreground pixels, for which almost all directions would be equally possible, so biasing the correct updating of the directions. The updated DM_G is shown in Fig. 2 right, where white pixels are those assigned to the background.

3 The Segmentation Method

Segmentation can be achieved by assigning to the background any pixel p whose score $M(p)$ in $DIFF$ is smaller than a threshold θ, which is set based on the directional features of the processed image. To this aim, we compute the arithmetic mean m of the scores $M(p)$ of all pixels that are currently foreground pixels, i.e., pixels that in DM_G are different from zero, and set θ equal to a given percentage of m.

Actually, we use two different percentages of m, so as to achieve two different values for θ, θ_u = $u \times m$ and θ_o = $o \times m$ with $u > o$, leading to two different segmented images, G_u and G_o. When the higher threshold value is adopted, only the pixels with higher probability to be true vessel pixels are assigned to the foreground in G_u. The resulting image is generally under-segmented, since pixels belonging to slightly lighter vessels or to capillaries may not survive thresholding. With the lower threshold value, also pixels with lower probability to be vessel pixels are selected, so achieving an over-segmented image G_o. The number of pixels that are not true vessel pixels is much smaller in G_u than in G_o. For the running example, the under-segmented

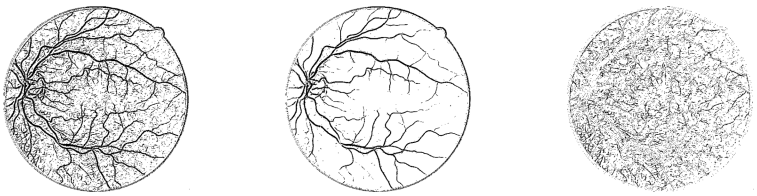

Fig. 3. The image G_u, left, the image G_o, middle, and the image difference $G_{o\text{-}u}$, right.

image G_u and the over-segmented image G_o, obtained for $u = 1.8$ and $o = 0.9$, are respectively shown in Fig. 3 left and middle. The image difference $G_{o\text{-}u}$ includes the pixels in the foreground of G_o but not in the foreground of G_u. See Fig. 3 right.

Linking and Cleaning

The image difference $G_{o\text{-}u}$ is rather noisy. In fact, the low threshold value $\theta\_o = o \times m$ adopted to obtain G_o guarantees the detection of almost all the true vessel pixels, but also causes the wrong assignment to the foreground of a number of false vessel pixels. Thus, we consider as necessary some cleaning of $G_{o\text{-}u}$, based on removal of small size components. Before doing the size-based cleaning, we link with each other small close components that we interpret as parts of vessels resulting in distinct connected foreground components of $G_{o\text{-}u}$ since some of their pixels were selected as foreground pixels when adopting both the lower and the higher threshold value.

For the linking process we consider components of $G_{o\text{-}u}$ with size smaller than a given maximum value *max*, since only small size components risk to be removed during the successive cleaning task. On the other hand, we do not consider components with very little size, say with size smaller than a given minimum value *min*, since these components most possibly consist of false vessel pixels. Thus, we consider for the linking process every component of $G_{o\text{-}u}$ whose size s is such that $min \leq s \leq max$. We have experimentally found that the better performance of the method is achieved in the average by setting $min = 16$ and $max = 150$. We use an iterated expansion process. At each iteration, the background neighbors of any pixel p in the components of $G_{o\text{-}u}$ selected for the linking process are assigned to the components provided that they are foreground pixels in G_o, and share with p the same direction. The first requirement guarantees that linking pixels were at least tentatively selected as vessel pixels in the over-segmented image; the second requirement guarantees that linking regards exclusively components that actually are part of vessels aligned along given directions. The number of iterations depends on the maximal distance between two components to be connected. Since G_o includes many false vessel pixels, we limit the number of iterations to two, which means that we can link to each other only components with a maximal distance of at most four pixels. The expansion process is followed by an iterated topological removal process, aimed at assigning again to the background all pixels added by the expansion process except those that favored linking of components. When no more pixels are removed, the surviving added pixels are assigned to the foreground in $G_{o\text{-}u}$.

Size-based cleaning is performed, by removing from $G_{o\text{-}u}$ all components with area smaller than a threshold σ. We have found that the best performance is obtained in the

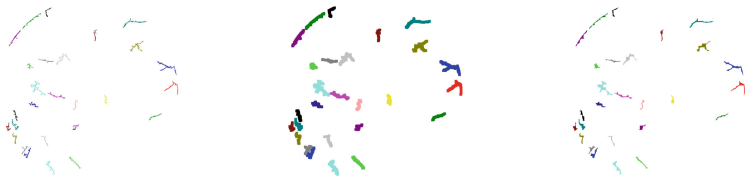

Fig. 4. The image $G_{o\text{-}u}$ after linking and cleaning, left, expansion, middle, and shrinking, right. (Color figure online)

average for $\sigma = 64$. The result obtained after linking and cleaning can be seen in Fig. 4 left. Different colors denote the identity labels of the components of $G_{o\text{-}u}$.

Selection of Foreground Components Consisting of True Vessel Pixels

We use the main feature characterizing vessels, i.e., the fact that vessels have linear structure, to distinguish in $G_{o\text{-}u}$ the components to be maintained in the foreground from those to be assigned to the background. We perform a small number of iterations of an expansion process, set to three in this work, during which thin holes and concavities possibly present in any component of $G_{o\text{-}u}$ are filled in, without creating an unwanted merging of foreground components. See Fig. 4 middle showing the resulting $G_{o\text{-}u}$ after the three iterations. Then, we perform an equal number of iterations of topological shrinking, during which pixels added to any component by the expansion process are removed provided that they have at least a neighbor in any other component, including the background. The resulting image $G_{o\text{-}u}$ is shown in Fig. 4 right. It can be observed that while components originally characterized by linear structure are not remarkably modified by the expansion/shrinking process as regards their size, components with a more complex structure, erroneously assigned to the foreground, have at the end of the process a significantly larger size. Actually, besides the changes in size, we also take into account the changes in maximal width of the components of $G_{o\text{-}u}$. This last feature is easily measured as the maximal value in the distance transform of each component before and after the expansion/shrinking process. Let W_{in} and W_{exp} be the initial maximal width and the maximal width after the expansion/shrinking process, respectively. Moreover, let A_{in} and A_{exp} be the initial area and the area after the expansion/shrinking process. We assign to the background any component for which it results $A_{exp}/A_{in} \geq 0.40$, or it is $0.30 \leq A_{exp}/A_{in} < 0.40$ and $W_{exp}/W_{in} \geq 2$. All other components remain in the foreground.

All components of $G_{o\text{-}u}$ surviving the expansion/shrinking process are recognized as consisting of vessel pixels. The pixels that belonged to these components before the expansion/shrinking process are transferred into G_u. The currently obtained G_u can be seen in Fig. 5 left.

Improving Segmentation

To recover missed vessel pixels, we apply to the components of G_u a slightly modified version of the process adopted to link close components of $G_{o\text{-}u}$. As done when processing $G_{o\text{-}u}$, background neighbors of pixels in components of G_u are added to the components only if they also exist in G_o. Differently from the process applied to $G_{o\text{-}u}$, the number of iterations is not fixed a priori; moreover, we have some tolerance on the direction of the pixels to be added. In detail, the expansion process terminates when no

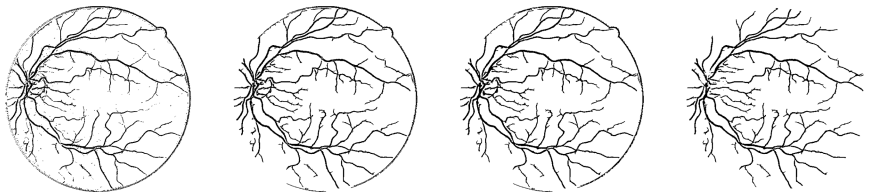

Fig. 5. The image G_u after adding the contribution of $G_{o\text{-}u}$, left; G_u after recovery of missed vessel pixels and removal of small components, middle left, G_u after foreground concavities and holes filling, middle right, and G_u after cleaning of the circular boundaries of retina and optical disc, right.

more pixels can be added; at each iteration, if d_i is the direction of a pixel p in a given component of G_u, any neighbor of p is parallelwise added to G_u, provided that its assigned direction is any out of $d_{i\text{-}1}$, d_i, d_{i+1}. Finally, differently from the process done on $G_{o\text{-}u}$, the expansion process is not followed by any iterated topological removal and we accept as vessel pixels all the recovered pixels. Finally, components of $G_{o\text{-}u}$ that notwithstanding the recovery process are still characterized by small size are removed. We use the same threshold $\sigma = 64$ already adopted for size-based cleaning of $G_{o\text{-}u}$. The resulting image is shown in Fig. 5 middle left.

By taking into account the average width and distance of the vessels in the DRIVE database, we found adequate to fill foreground concavities up to 2-pixel wide. Since the concavities filling process is the same used in [12], for space limitation we do not describe again it here. Very small size holes (consisting of at most 16 pixels in this work) are also filled in. The resulting image G_u is shown in Fig. 5 middle right.

The last process is devoted to cleaning of the circular boundaries of retina and of optical disc. The masks available in the DRIVE database are used to extract only the circular part of G_u corresponding to the retina. However, some pixels in proximity of the boundary of the circular mask may have been erroneously interpreted as vessel pixels. Thus, we dilate the circular boundary of the mask by means of a structuring element of radius 20. Foreground pixels reached by the dilation process are marked as removable. Then, an iterated un-marking process is accomplished that removes the marker from any neighbor q of an un-marked vessel pixel p with direction d_i if the direction of q is compatible with the direction of p, i.e., the direction of q is any out of $d_{i\text{-}1}$, d_i, d_{i+1}. The number of iterations of the un-marking process is equal to 20, i.e., to the radius of the structuring element. Pixels that at the end of the un-marking process are still marked are assigned to the background.

To remove pixels erroneously detected as vessel pixels within the optical disc, we first identify in G a region of interest, ROI, that includes the optical disc. It can be noted that the optical disc includes pixels with remarkably lighter gray-levels, and its maximal diameter and position are predictable in the DRIVE database. Thus, we use a sliding window of size 121×121 which swipes the image in the rectangular area where the optical disc is expected to be located. The pixels in the sliding window with gray-level larger than 80% of the maximal gray-level in G are counted while the window swipes the image. The position of the window in correspondence of which the number of counted pixels has the highest value defines the ROI. We observe that some

vessels exist in the optical disc and these are characterized by rather dark gray-level and almost horizontal direction. Thus, out of all pixels of the ROI that have been assigned to the foreground of G_u, we assign to the background those with gray-level larger than a suitable percentage, *perc*, of the arithmetic mean of the gray-levels within the ROI and with direction i which is not compatible with the horizontal direction. We have obtained the best performance by setting *perc* = 90% and by considering compatible with the horizontal direction d_i, characterized by $i = 7$, any direction out of d_5, d_6, d_7, d_8, and d_9. The final segmentation result is shown in Fig. 5 right.

4 Experimental Results and Concluding Remarks

The suggested segmentation method has been checked on the 40 images of the DRIVE database, by using as ground truth the manually segmented images included in the database. Actually, two manually segmented images are available for each of the 20 retinal images forming the test set, and one manual segmentation for the remaining 20 images forming the training set. Since our segmentation method is unsupervised, we have not done any difference between the test set and the training set.

For a qualitative evaluation of the method, see Fig. 6, where the results for the four images 09_test, 19_test, 27_training, and 32_training are shown.

To quantitatively evaluate the method, let TP and TN count the pixels correctly classified as vessel pixels and as non-vessel pixels, and FP and FN the pixels incorrectly classified as vessel pixels and as non-vessel pixels, and compute:

Accuracy = (TP + TN)/(TP + FN + TN + FP)
Sensitivity = TP/(TP + FN)
Specificity = TN/(TN + FP)

The average values of Accuracy, Sensitivity and Specificity have been computed for the 20 test images with respect to the first ground truth to compare the performance of our method with that of other 6 unsupervised segmentation methods in the literature,

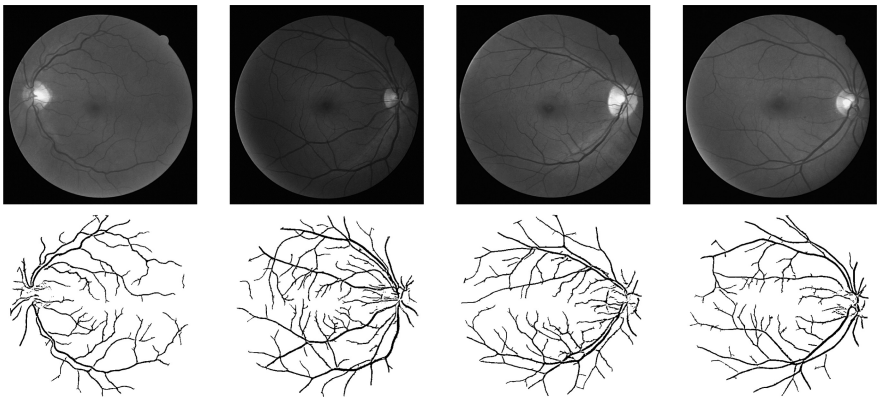

Fig. 6. Some examples, top, and the segmentation results, bottom.

Table 1. Performance comparisons.

	Accuracy	Specificity	Sensitivity
Mendonca and Campilho [8]	0.946	0.976	0.734
Al-Rawi et al. [17]	0.953		
Zhang et al. [18]	0.938	0.972	0.712
Zhao et al. [19]	0.954	0.982	0.742
Azzopardi et al. [11]	0.944	0.970	0.765
Frucci et al. [12]	0.955	0.985	0.640
Our method	0.956	0.985	0.660

[8, 11, 12, 17–19]. See Table 1. Our method is slightly better as regards Accuracy and Specificity, and inferior for Sensitivity, but generally produces results qualitatively more satisfactory.

References

1. Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Trans. Med. Imaging **23**, 501–509 (2004)
2. Teng, T., Lefley, M., Claremont, D.: Progress towards automated diabetic ocular screening: a review of image analysis and intelligent systems for diabetic retinopathy. Med. Biol. Eng. Comput. **40**, 2–13 (2002)
3. Haddouche, A., Adel, M., Rasigni, M., Conrath, J., Bourennane, S.: Detection of the foveal avascular zone on retinal angiograms using Markov random fields. Digit. Sig. Process. **20**, 149–154 (2010)
4. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Trans. Circ. Syst. Video Technol. **14–1**, 4–20 (2004)
5. Li, Q., You, J., Zhang, D.: Vessel segmentation and width estimation in retinal images using multiscale production of matched filter responses. Expert Syst. Appl. **39**(9), 7600–7610 (2012)
6. Rangayyan, R., Zhu, X., Ayres, F., Ells, A.: Detection of the optic nerve head in fundus images of the retina with Gabor filters and phase portrait analysis. J. Digit. Imaging **23**, 438–453 (2010)
7. Wang, Y., Ji, G., Lin, P., Trucco, E.: Retinal vessel segmentation using multiwavelet kernels and multiscale hierarchical decomposition. Pattern Recogn. **46–8**, 2117–2133 (2013)
8. Mendonça, A.M., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Trans. Med. Imaging **25**(9), 1200–1213 (2006)
9. Fraz, M.M., Barman, S.A., Remagnino, P., Hoppe, A., Basit, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G.: An approach to localize the retinal blood vessels using bit planes and centerline detection. Comput. Methods Programs Biomed. **108**(2), 600–616 (2012)
10. Ricci, E., Perfetti, P.: Retinal blood vessel segmentation using line operators and support vector classification. IEEE Trans. Med. Imaging **26**(10), 1357–1365 (2007)

11. Azzopardi, G., Strisciuglio, N., Vento, M., Petkov, N.: Trainable COSFIRE filters for vessel delineation with application to retinal images. Med. Image Anal. **19**(1), 46–57 (2015)
12. Frucci, M., Riccio, D., Sanniti di Baja, G., Serino, L.: Severe, segmenting vessels in retina images. Pattern Recogn. Lett. **82**, 162–169 (2016)
13. Sun, K., Chen, Z., Jiang, S., Wang, Y.: Morphological multiscale enhancement, fuzzy filter and watershed for vascular tree extraction in angiogram. J. Med. Syst. **35**(5), 811–824 (2010)
14. Imani, E., Javidi, M., Pourreza, H.R.: Improvement of retinal blood vessel detection using morphological component analysis. Comput. Methods Programs Biomed. **118**(3), 263–279 (2015)
15. Frucci, M., Riccio, D., di Baja, G.S., Serino, L.: Using contrast and directional information for retina vessels segmentation. In: Proceedings of the SITIS 2014, pp. 592–597. IEEE CS (2014)
16. Zhao, Y.Q., Wang, X.H., Wang, X.F., Shih, F.Y.: Retinal vessels segmentation based on level set and region growing. Pattern Recogn. **47**(7), 2437–2446 (2014)
17. Al-Rawi, M., Qutaishat, M., Arrar, M.: An improved matched filter for blood vessel detection of digital retinal images. Comput. Biol. Med. **37**, 262–267 (2007)
18. Zhang, B., Zhang, L., Zhang, L., Karray, F.: Retinal vessel extraction by matched filter with first-order derivative of Gaussian. Comput. Biol. Med. **40**(4), 438–445 (2010)
19. Zhao, Y., Rada, L., Chen, K., Harding, S.P., Zheng, Y.: Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. IEEE Trans. Med. Imaging **34**(9), 1797–1807 (2015)