Journal of Microscopy, Vol. 234, Pt 2 2009, pp. 147–157
Received 15 September 2008; accepted 5 January 2009

Fast linear feature detection using multiple directional non-maximum suppression

C. SUN & P. VALLOTTON
CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, New South Wales 1670, Australia

Key words. Gap linking, linear feature detection, multiple directional non-maximum suppression, multiple local maxima.

Summary

The capacity to detect linear features is central to image analysis, computer vision and pattern recognition and has practical applications in areas such as neurite outgrowth detection, retinal vessel extraction, skin hair removal, plant root analysis and road detection. Linear feature detection often represents the starting point for image segmentation and image interpretation. In this paper, we present a new algorithm for linear feature detection using multiple directional non-maximum suppression with symmetry checking and gap linking. Given its low computational complexity, the algorithm is very fast. We show in several examples that it performs very well in terms of both sensitivity and continuity of detected linear features.

Introduction

Detecting linear features is important in applications such as retinal vessel extraction, fingerprint analysis and skin hair removal for melanoma detection in the medical and biometrics areas; in applications such as neurite outgrowth (Meijering *et al.*, 2003; Ramm *et al.*, 2003; Van de Wouwer *et al.*, 2004), tree bark, tree branches, plant roots and leaf vein/skeleton detection in areas related to biological samples and in applications such as road crack and roads and valleys detection in the infrastructure area (Fischler *et al.*, 1981).

A number of techniques exist in the literature for linear feature detection. Many algorithms have been developed specifically for the detection of retinal vessels (see Kirbas & Quek, 2004, for a recent review). One approach is based on a series of directional filters corresponding to the direction of the structures in the image (Bamberger & Smith, 1992). In this context, steerable filters (Jacob & Unser, 2004), 2D matched filters (Chaudhuri *et al.*, 1989), maximum gradient profiles (Colchester *et al.*, 1990), fiberscoring (Lichtenstein *et al.*, 2003) and directional morphological filtering (Soille & Talbot, 2001) have been used. Such techniques can be termed 'template-based' or 'model-based' and tend to be slow, as they amount to calculating a correlation coefficient at every position in the image and for each orientation considered. Another approach uses the classical gradient/curvature or the Hessian-based detectors. This includes thin nets or crest lines (Monga *et al.*, 1997) and ridges (Gauch & Pizer, 1993; Eberly, 1996; Lang *et al.*, 1997).

Tracking techniques, including stick growing, have also been used (Nelson, 1994; Can *et al.*, 1999; Tolias & Panas, 1998). Tracking-based approach requires initial locations along linear features. Often, this amounts to user intervention. Methods using edge operators to detect pairs of edges and graph-searching techniques to find the centrelines of vessel segments are presented in Fleagle *et al.*, (1989) and Sonka *et al.*, (1995). Related techniques that are edge- or 'roof'-based also exist (Nevatia & Babu, 1980; Zhou *et al.*, 1989; Steger, 1998). Canny's ridge or roof detector requires the estimation of the direction of the ridge or roof first, which is not an easy task for linear features (Canny, 1983). Yet other algorithms find linear features by classification using a neural network scheme with supervised training (Staal *et al.*, 2004), S-Gabor filter and deformable splines (Klein *et al.*, 1997) and mathematical morphology (Zana & Klein, 2001).

In this paper, we propose a new fast algorithm for linear feature detection using multiple directional non-maximum suppression (MDNMS). The rationale for the method is that when a linear window (see Fig. 1) crosses a linear feature, the intensity along the linear window presents a maximum. This maximum position is marked as a linear feature candidate. All other pixels within the linear window can be discarded. Because identifying a maximum value amongst a few pixels is a computationally cheap process, our approach for linear feature detection is very fast and should be very useful in many real-time, interactive or high-throughput applications.

Correspondence to: C. Sun. Tel: +61-2-9325-3207; fax: +61-2-9325-3200; e-mail: changming.sun@csiro.au

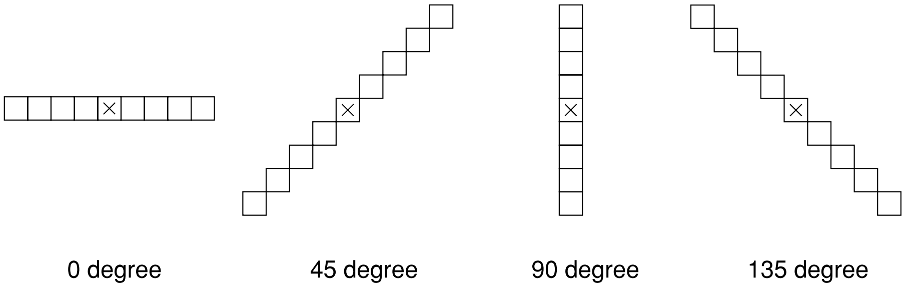

Fig. 1. Illustration of linear windows at four different directions. '×' indicates the centre of the linear windows. The length of the linear window shown is 9 pixels.

Multiple directional non-maximum suppression

Linear features are defined as a connected sequence of points along which the image presents an intensity maximum in the direction of the largest variance, gradient or surface curvature (i.e. perpendicular to the linear feature). This direction may be obtained by the use of the computationally expensive Hessian-based detectors or matched filters. However, direct use of image gradient on linear features is not very reliable. Rather than searching for the local direction of linear features, we exploit the definition of linear features directly and perform non-maximum suppression (NMS) in multiple directions to identify candidate pixels on a linear feature.

Non-maximum suppression is the process of marking all pixels whose intensity is not maximal within a certain local neighbourhood as zero. The shape of this local neighbourhood is usually a square or a rectangular window. We choose this local neighbourhood as a linear window in this paper. A linear window can be oriented in different directions. A directional local maximum is a pixel not surrounded by pixels of higher grey values in a linear window, and they are marked as one in our algorithm. There are a number of approaches that one can use to actually obtain the directional local maximum within a 1D window, as described in Breen & Jones, (1996) and Neubeck & Van Gool, (2006). A local maximum can also be obtained using basic morphological operators by checking whether a pixel value in the input image is the same as that in the dilated image. We use a simple and yet fast algorithm for finding directional local maximum at one particular angle, as described in the algorithm below, in which 'in' and 'out' are the input and output images, respectively.

```
Procedure Directional Local Maximum():
for each pixel $i$ in image do
    if out[$i$] $\neq$ Processed then
        for each pixel $j$ ($\neq i$) in linear window do
            if in[$i$] $\leq$ in[$j$] then
                go to jump;
            else
                out[$j$] = Processed
        out[$i$] = in[$i$]
    jump:
```

The '≤' used in the pseudo code enables a program to detect strictly a single maximum point. If two points with equal value are in the same window, there will be no output. If '<' is used in the pseudo code, all the points in a flat region will be detected. Note that the code above is just for one direction. For a different direction, the pixels in the linear window will be different.

Figure 1 shows four examples of linear windows at angles of 0°, 45°, 90° and 135°. Additional directions, such as 22.5°, 67.5°, 112.5° and 157.5°, can also be used. Non-maximum suppression is performed successively for each direction and at each pixel. The result is independent of the order of the scanning process. The longer the linear window is, the lower the number of candidate pixels will be detected in an image. The ideal scenario is when the linear window crosses the linear feature at right angle, as this will produce maximum contrast.

Figure 2 shows example responses of directional NMS for four window directions. Figure 2(a) is an input image of neurons and their associated neurites. Figures 2(b)–(e) are the directional NMS responses at angles 0°, 45°, 90° and 135°. The outputs of the directional NMS are binary images, as opposed to grey-scale images produced by most directional filter methods. This eliminates the need for a mostly arbitrary thresholding step.

Fast linear feature detection

In this section, we outline the main steps of our fast algorithm for obtaining linear features from images.

Combining multiple directional non-maximum suppression

We use the union of MDNMS responses, as given in Eq. (1), for linear feature detection, because linear features in an image can be oriented in any direction:

$$\mathtt{L} = \bigcup_{i=1}^{N_D} \mathtt{L}_{D_i}, \qquad (1)$$

where $\mathtt{L}_{D_i}$ is the result for NMS in direction D_i, N_D is the number of directions used and $\mathtt{L}$ is the result image. For each direction or angle for the linear window, the directional local maximum for L_{D_i} can be obtained using the algorithm described in the

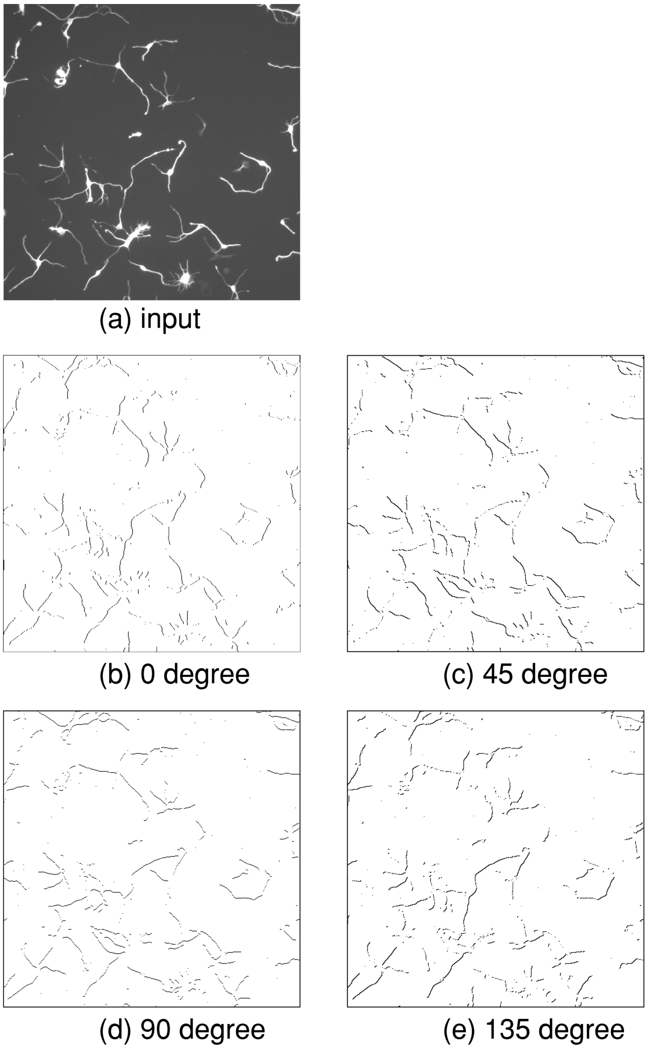

Fig. 2. Non-maximum suppression responses for four linear windows in different directions, with the length of the linear window being 11 pixels. (a) Input image. (b–e) NMS responses at angles 0°, 45°, 90° and 135°.

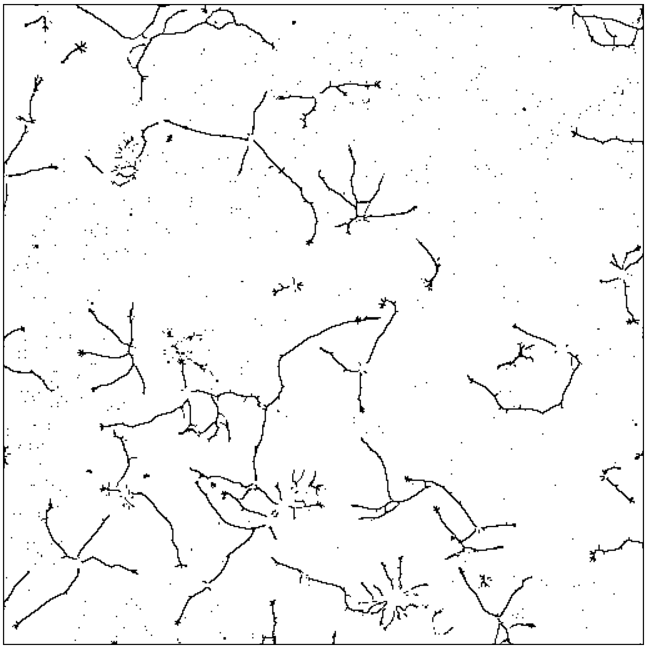

Fig. 3. Combined response of the non-maximum suppression as shown in Figs. 2(b)–(e).

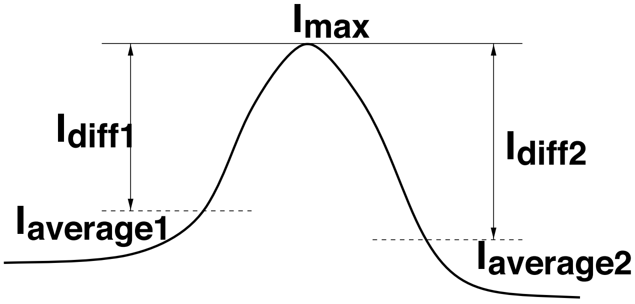

Fig. 4. Symmetry profile of a cross-section of a linear feature.

previous section. The number of directions used N_D can be either 4 or 8, or even 2. Figure 3 shows the union result of the multiple NMS responses given in Figs. 2(b)–(e). It can be seen from Fig. 3 that all of the major linear features present in Fig. 2(a) are found.

Symmetry check

True linear features are characterized by an approximately symmetric intensity profile across the feature, as opposed to edges, which would show a roughly step profile. This translates into approximately equal intensity values around the central pixel on both sides of the linear window. Figure 4 illustrates the parameters that characterize the shape of a profile across a linear feature. I_{max} is the value of a local directional maximum at the central pixel of the linear window. I_{average1} and I_{average2} are the average intensity values over the two sides of the local maximum within the local window, and I_{diff1} and I_{diff2} are the differences between the maximum value and these two average values. For a linear feature in the image, both I_{diff1} and I_{diff2} should be large. If only one of these values is larger, the local maximum is most likely an edge rather than a linear feature. The parameter I_{diff1} or I_{diff2} can be used to control the sensitivity of the algorithm. Lowering the value of I_{diff} can increase the number of linear features detected. In practice, one needs to adjust the parameter so that satisfactory feature detection can be achieved.

Extending to multiple local maxima

In some images, linear features may be very close to each other in some areas. The NMS process described above may detect only one of the two or more linear features in close proximity. Figure 5 shows an example in which there are

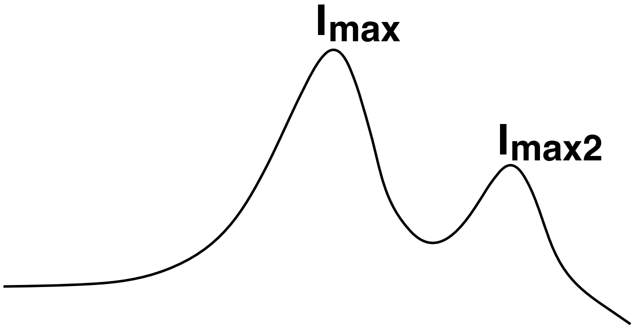

Fig. 5. Multiple local maxima within a linear window.

two local maxima within the linear window. We can extend the NMS process so that multiple local maxima are detected for any particular linear window. Once a local maximum is found at the centre of a linear window, then a second pass of local maximum searching is carried out using a smaller linear window size within the original linear window. Because of the use of a smaller window size, other local maxima may be found within the original window. For the newly found local maxima, the highest one can be chosen as the second local maximum in the original window. The selection of a second local maximum or more local maxima within the same window enables us to detect multiple, closely spaced linear features. Figure 5 shows a second local maximum with intensity $I_{\max2}$. We also check that the intensity $I_{\max2}$ is not very different from $I_{\max}$, and that the symmetry condition is also similar.

Removing small objects

The union of the multiple responses of NMS in different directions generate some small objects that may not belong to linear features. We fit an ellipse to each objects and remove the objects if the major axis of the best-fitted ellipse is small. A simple pixel count can also be used, and it will work most of the time. One may also use both ellipse-fitting parameters and the area of an object for small object removal. If the area of an object is very small, then it is removed. If the area of an object is very large, then it will be kept. In these two cases, no ellipse fitting is necessary. For those objects with sizes in between, ellipse-fitting parameters will be used to decide whether to remove the object. Because we are to detect linear features which should be thin and long objects, a thin and long object is preferred to a short cluster of features, even though they may have the same number of pixels. Using the fitted ellipse information, we will have more control on the shape of a linear feature.

Linking broken linear features

There may be small gaps in the combined responses of NMS that break the continuity of the linear features when the signals of the linear features are weak. We restore the continuity of weak parts of linear features by joining the endpoints to neighbouring features through the shortest path. The connection of an endpoint to neighbouring features can take the form of endpoint-to-endpoint connection or endpoint-to-skeleton connection. The endpoints are detected using standard mathematical morphology algorithms (Serra, 1982 p. 392) from the skeletons of the combined NMS responses. The shortest path linking step ensures that the connections follow the ridges of the linear features. The computational overhead remains small because the operation is only performed on small gaps.

For finding the path, one can choose to use a traditional shortest path method or a polar transformation method. The advantage of the traditional method is that one can have paths that have more complex shape. In both cases, we will need to use some kind of average or normalized path value to determine which path to choose. We will use the polar transformation method for path linking because it is easier to obtain the average path cost, even though there is an added cost for the transformation step. Because the links are going to be short, it will probably not make much difference as to which method is used.

Figure 6 illustrates the process of gap linking from an endpoint to the features in its neighbourhood. Figure 6(a) is a sub-region of an input image. Figure 6(b) shows an endpoint in red and its neighbourhood is shown with a green circle. The disc region bounded by the green circle is transformed to the polar coordinate system around the centre. The transformed image (as shown in Fig. 7c) will be used to find the best path to achieve gap linking. Figure 6(c) shows linear features with gaps linked.

Figure 7 shows the process of using the shortest path technique (sect. II of Sun & Pallottino, 2003) to link endpoints and nearby linear features. Figure 7(a) shows the average distance cost from the top to the bottom of the transformed image or from the centre to the edge of the disc, with blue indicating smaller cost and red indicating bigger cost.

Figure 7(b) shows the transformed image with feature skeletons in the neighbourhood. Figure 7(c) shows the best path obtained (from the blue region to the top of the image, as shown in Fig. 7a).

The algorithm steps for using the shortest path technique to link endpoints to linear features are:

(1) Start from an endpoint.
(2) Connect this endpoint to linear features in the neighbourhood:
  (a) Convert a local circular region around an endpoint for both the input intensity image and the image with linear features into polar coordinates and form images A and B.
  (b) Obtain the averaged distance cost of image A (Sect. II of Sun & Pallottino, 2003 sect. II).

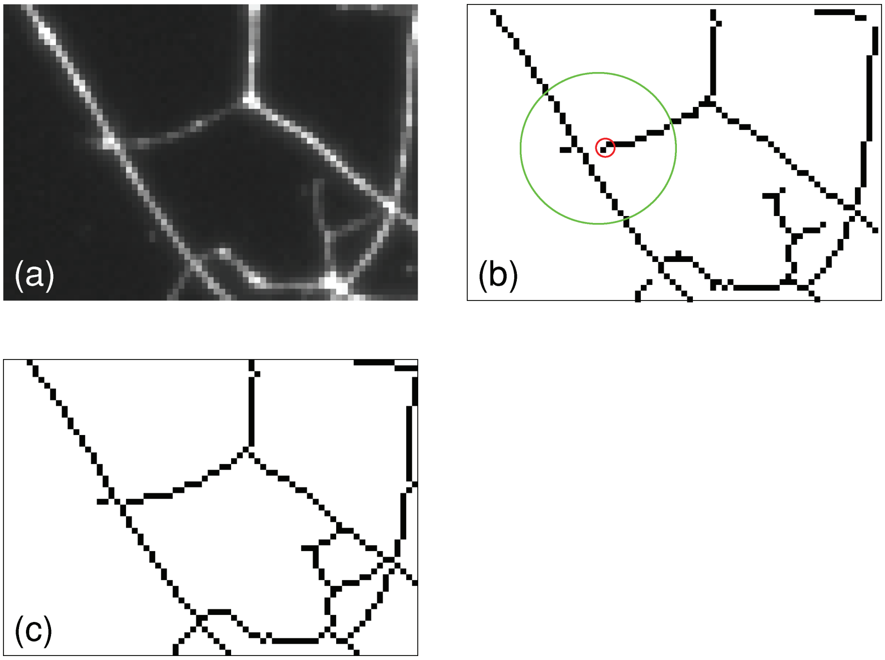

Fig. 6. Endpoint and its neighbourhood for gaps linking. (a) Sub-region of input image. (b) Endpoint and its neighbourhood shown with red and green circles. (c) Skeleton with gaps linked.

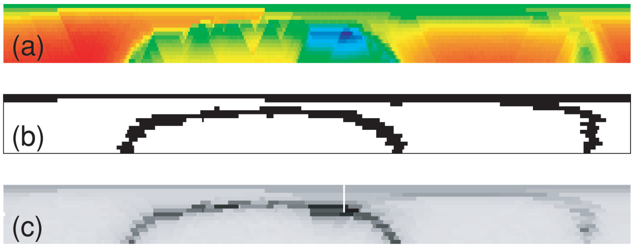

Fig. 7. Shortest path from endpoint to skeletons. (a) Distance map in the polar coordinate system. (b) The skeletons in the neighbourhood of an endpoint. (c) Shortest path extracted (the short segment in white) from endpoint to a skeleton in the neighbourhood of the endpoint.

(c) Find a pixel *P* in the averaged distance cost image that has the minimum value, and at the same position in image *B*, there exists a linear feature.
(d) Find the shortest path from *P* to the top of the image *A*. The average image intensity along this path must be above a certain threshold.
(e) Convert back to Cartesian coordinate system for the path.

Extending to 3D images

The algorithms for linear feature detection using NMS can be easily extended for 3D images using 3D linear windows, as shown in Fig. 8. We used either 3 or 9 directions, although additional directions could also be used for obtaining the MDNMS.

Algorithm steps

The algorithm steps for linear feature detection are summarized here:

(1) Invert the image intensity if necessary.
(2) Detect areas of low variance (for background extraction) if necessary and perform local smoothing on whole image.
(3) Carry out MDNMS and take the union of the 4 or 8 (3 or 9 for 3D images) NMS outputs. The symmetry check is performed during the process of NMS. Optionally, multiple local maxima are identified within a linear window.
(4) Remove small objects and carry out thinning of the objects.
(5) Obtain endpoints and link the endpoints of linear features to neighbouring features.

Experimental results

This section presents experimental results of our algorithm obtained in a variety of images. The results obtained using the ranked filter algorithm (Soille & Talbot, 2001) are shown for comparison purpose.

Neurite outgrowth detection

The capacity to measure neurite outgrowths is central to discovering and profiling new drugs against neurodegenerative diseases. Our algorithm has been tested for neurite outgrowths detection on a set of 93 images of neurons using a single set of parameters. Figure 9 shows the results for a representative image. Figure 9(a) shows the

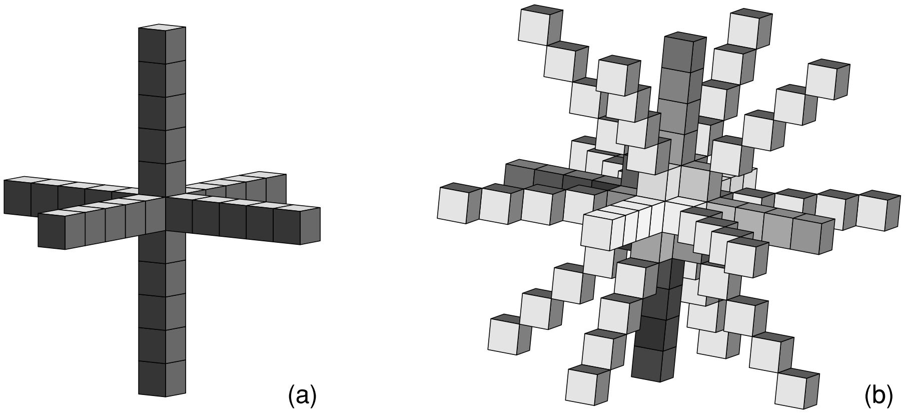

Fig. 8. Illustration of several linear windows in different directions. The length of the linear window shown is 9 pixels. (a) Three linear windows together. (b) Nine linear windows together.

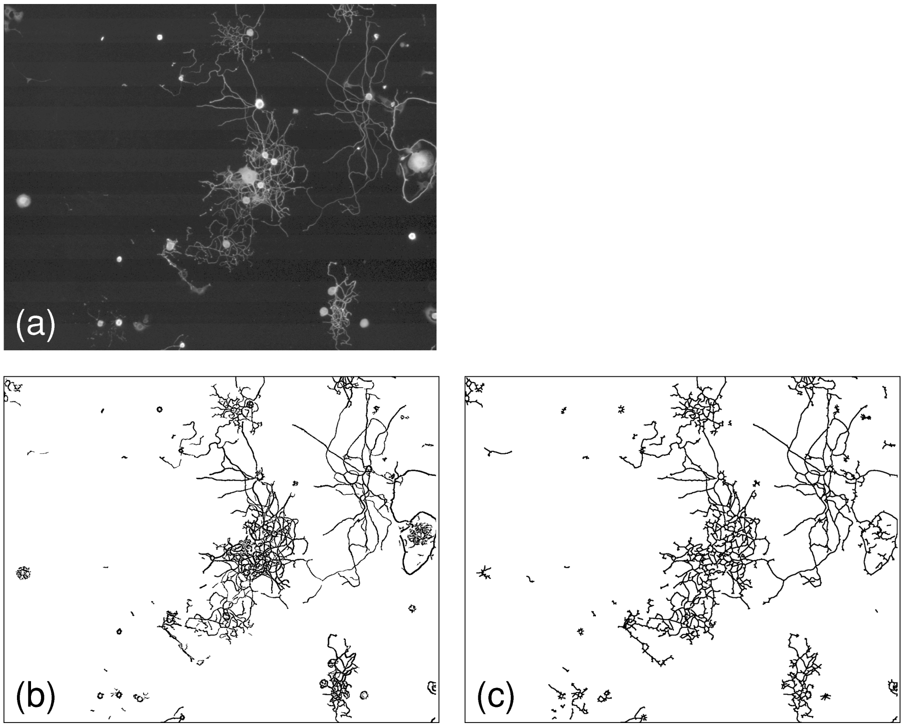

Fig. 9. Example of neurite outgrowth detection. (a) Input image. (b) Result of the ranked filter approach. (c) Result of our algorithm.

input image, Fig. 9(b) shows the result obtained from the ranked filter approach for comparison and Fig. 9(c) shows the result obtained from our algorithm. Our MDNMS algorithm delivered better connectivity for linear features and generated thinner skeletons of the linear features. It was also more than eight times faster than the ranked filter approach.

Other applications

Image analysis of sub-cellular structures has become a key tool for drug discovery, proteome characterization and automated functional analysis (Boland & Murphy, 2001). Figure 10 shows how our algorithm identified the major actin filaments. Lowering the parameter I_{diff} in our algorithm can increase the number of filaments detected.

Figure 11 demonstrates the performance of our algorithm for wrinkle detection, hair detection in an image with skin melanoma and fingerprint analysis. Some very short line segments shown in the result image that are perpendicular to the true fingerprint ridges are due to the low quality of the fingerprint image and lack of pre-processing.

Linear feature detection was also performed on images of leaf veins, onion layers, patterns on butterfly, insect wings and sperms, as shown in Fig. 12. Tests on a number of other

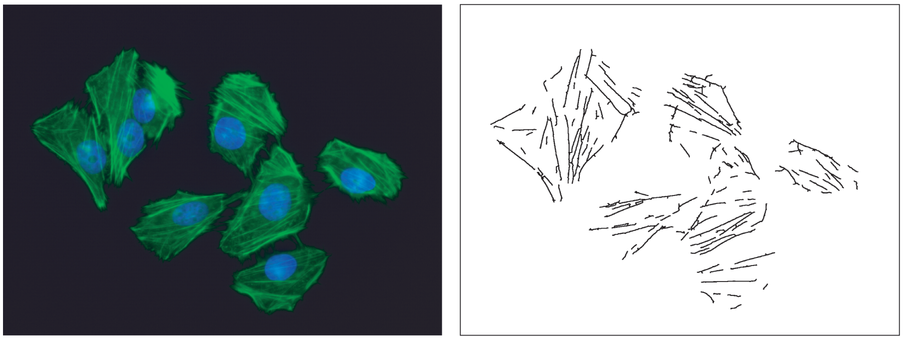

Fig. 10. Example of compartment assay of actin filaments (input image and the linear features detected).

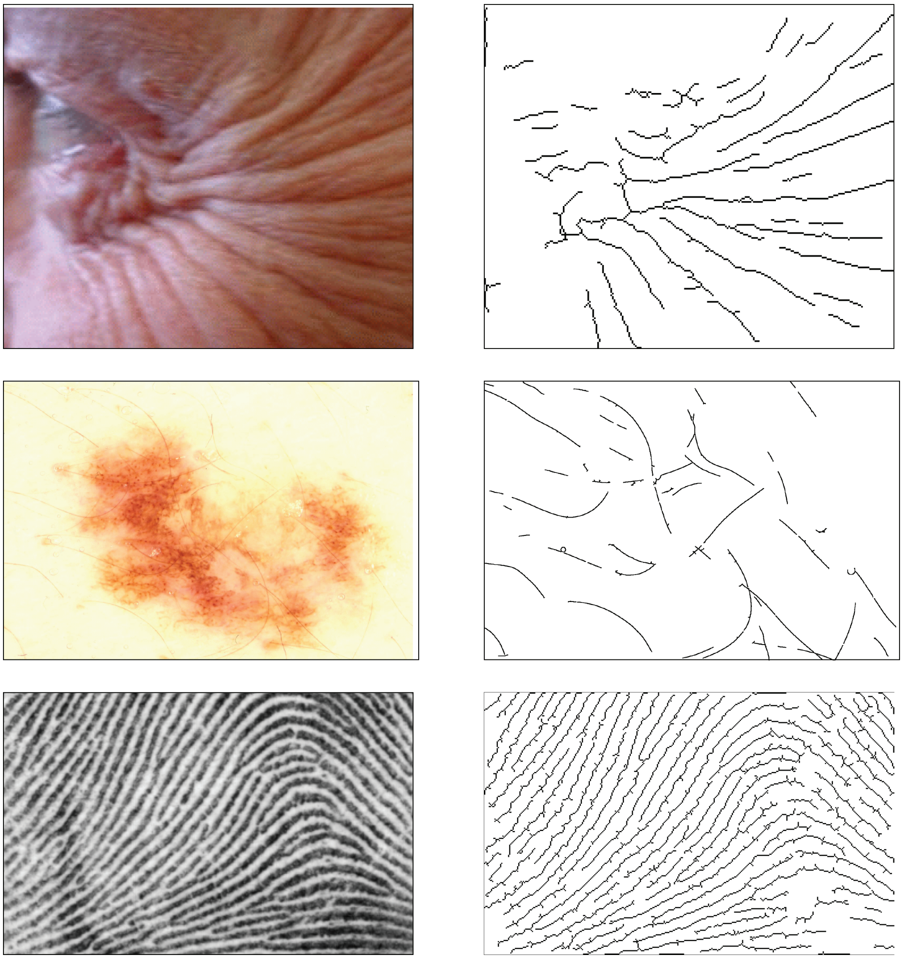

Fig. 11. Examples of medical applications.

applications have also been carried out, such as cracking in old paintings, patterns on a tile, surface scratches, lightening bolts, Chinese characters, features on digital elevation models, roads in an aerial image, cracks on the ground and brick patterns. The results show that our algorithm is effective.

Figure 13 shows the result for 3D images. Figure 13(a) is the transparent view of the input image, and Fig. 13(b) is one view of the detected 3D linear features.

CPU time

We also tested the CPU time of our algorithm. The computer used was an Intel Pentium 4 (Dell, Round Rock, Texas, USA), with a 2.66 GHz CPU under the Linux operating system, and our images had 8 bits/pixel. For colour images, only the intensity information or a single channel was used. Table 1 shows the CPU times (including image I/O) to compare our

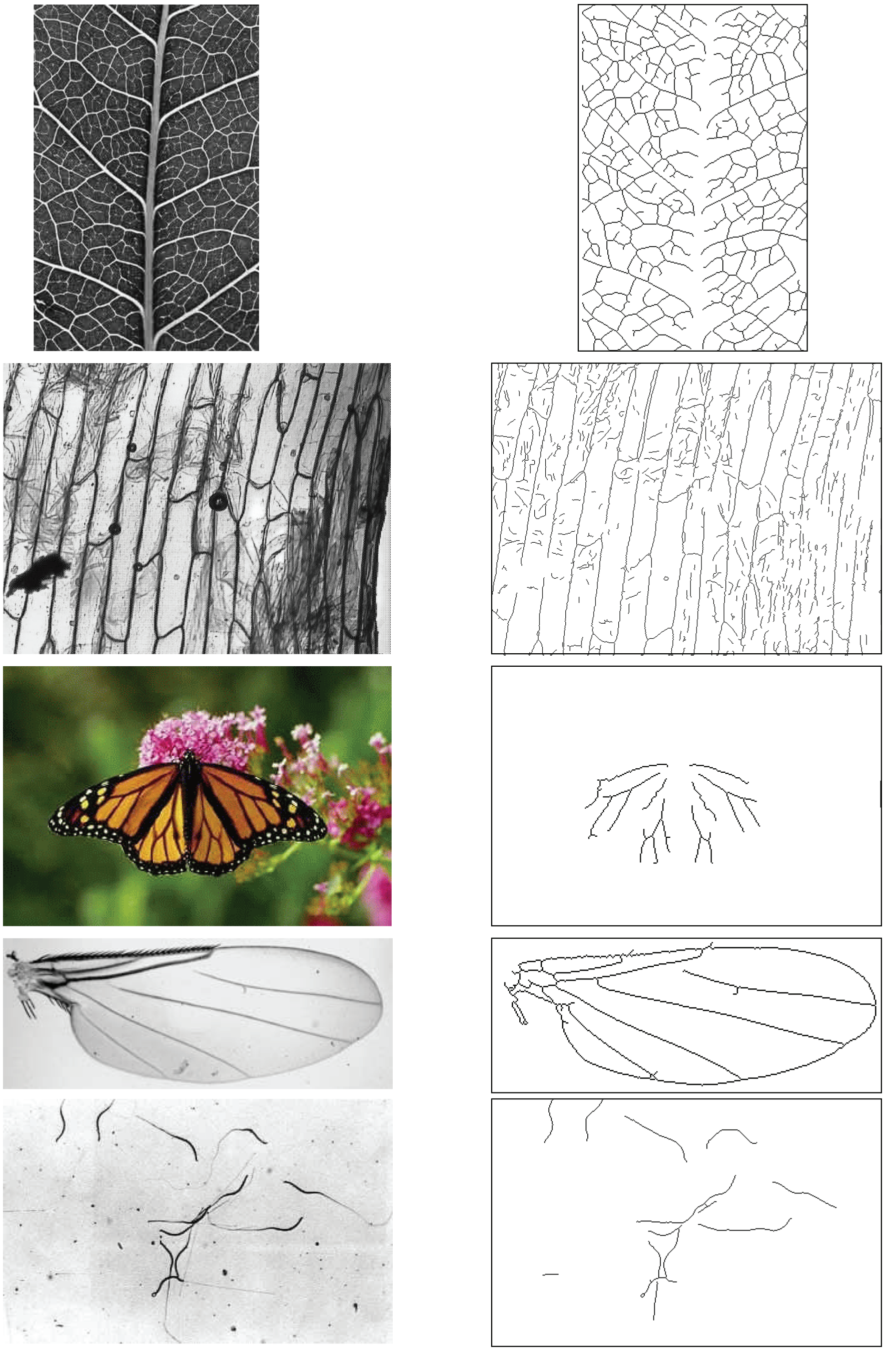

Fig. 12. Examples of biological applications.

method with the ranked filter approach in different sizes of images. Our new algorithm is more than eight times faster than the latter.

Figure 14 shows linear feature detection results using various numbers of orientations for the linear window. It can be seen from Fig. 14 that the use of more orientations gives better continuity of features. Our experience, however, is that little improvement is obtained beyond four orientations. Therefore, almost all results in this contribution were produced using four linear window orientations. Table 2 shows the running times for our algorithm with increasing number of orientations.

5 Discussion

Because of its simplicity, our MDNMS algorithm is very fast and should be useful in many real-time, interactive or high-throughput applications. Our method also demonstrates very

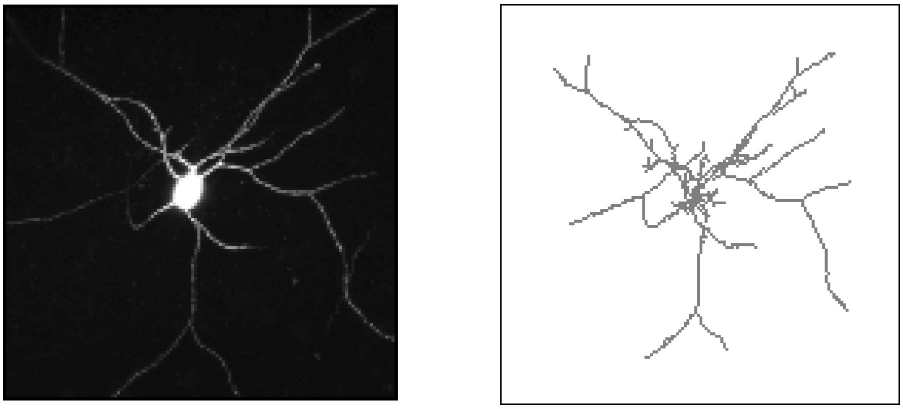

Fig. 13. Transparent view of the input 3D image and one view of the detected 3D neurite features.

Table 1. CPU times of the ranked filter algorithm and our MDNMS algorithm.

	CPU times (s)	
Image size	Rank filter	Our algorithm
512 × 512	1.41	0.17
1336 × 1206	8.80	0.84

Table 2. CPU times of our MDNMS algorithm with different number of linear window orientations.

	CPU times (s) with different number of orientations		
Image size	**2**	**4**	**8**
512 × 512	0.12	0.17	0.27
1280 × 1024	0.38	0.50	0.68

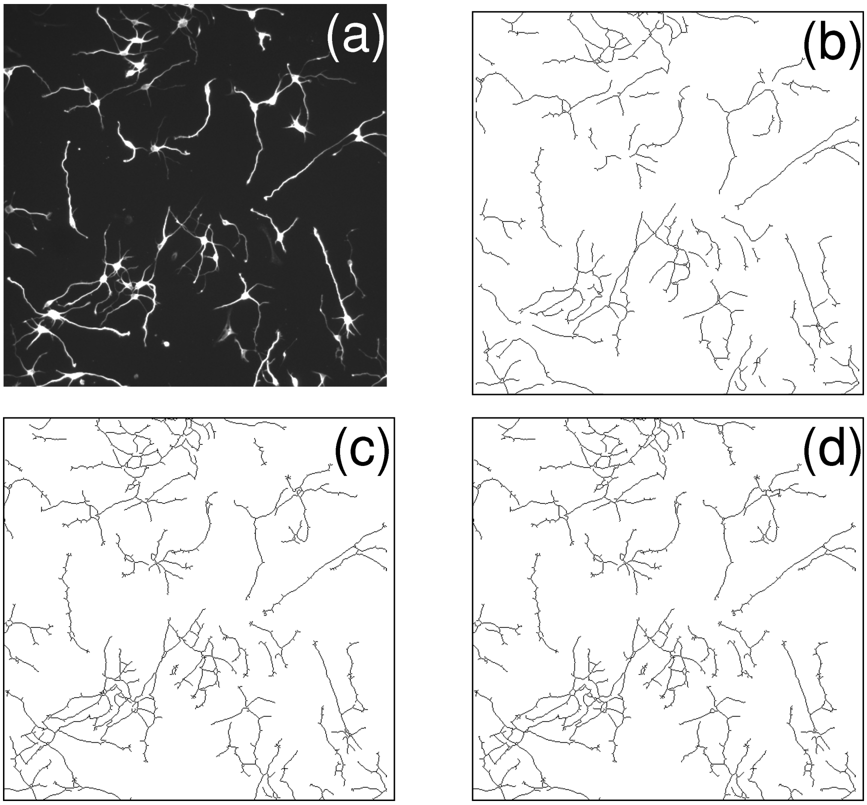

Fig. 14. Linear feature detection using different number of orientations. (a) Input image. Linear feature detection results with different number of linear windows (b) 2, (c) 4 and (d) 8.

good feature detection results. Only a few parameters are necessary for optimizing the response: the linear window size, linear feature strength and the maximum gap size. The method should appeal to users in a wide range of communities. One can also note that the algorithm is robust to background variation, as the main operation, NMS, is a local operation.

To increase the processing speed, an estimate of the local variance, with a square window at each point in the image, can be computed very quickly using recursive techniques (Sun, 2001). Those image positions with low local variance are then masked out, as they are unlikely to belong to a linear feature. Only those foreground pixels are processed for NMS. In extreme cases, two neighbouring pixels on a linear feature can have the same maximum value in the linear window. The likelihood of this condition can be decreased by first smoothing the image with a small window. A size of 3×3 has been found useful in terms of processing speed and preservation of weak linear features.

If feature boundaries are needed, extra steps are necessary. Boundaries can be used for width measurements. We can use the shortest path technique or multiple-path techniques for such purposes (Sun & Appleton, 2005; Lagerstrom *et al.*, 2008; Sun *et al.*, 2009).

The main focus of our approach is on linear feature detection rather than on image or feature enhancement. Our algorithm is probably more sensitive to noise than the kernel-based algorithms using 2D cross-correlations. Our approach will produce better results if the input image is pre-processed with feature enhancement procedures such as anisotropic filtering or by applying matched filters. Another feature of our approach is that it is easy to extend to 3D images, especially the NMS stage, which takes the majority of the processing time. Currently, the algorithm does not use multi-scale techniques. In certain applications, there might be linear features that occur at multiple scales. In such cases, multiple-scale techniques should be considered (Ter Haar Romeny, 2003), with lower resolution for detecting thick features and higher resolution for detecting thinner features.

Conclusions

We have developed a new algorithm for fast linear feature detection based on MDNMS with symmetry checking and gap linking. A comparison was carried out with the ranked filter approach and very favourable results were obtained in terms of both quality and speed. The new algorithm can be used in many application areas and a wide range of results have been shown.

Acknowledgements

We are grateful to the reviewers for valuable suggestions that greatly improved the paper. We thank Dr. Michael Buckley, Dr. Hugues Talbot, Ms. Leanne Bischof and Mr. Ryan Lagerstrom at CSIRO Mathematical and Information Sciences, Australia, for the implementation of the ranked filter approach. We are grateful to Dr. Luke Domanski for suggestions in speeding up the polar transformation algorithm. We thank Dr. Dadong Wang of CSIRO for his comments on the paper. Input data for Fig. 13 are courtesy of Professor P. Sah of the Queensland Brain Institute, Australia. The preliminary conference version of this paper was presented in Sun & Vallotton, (2006).

References

Bamberger, R.H. & Smith, M.J.T. (1992) A filter bank for the directional decomposition of images: theory and design. *IEEE Trans. Signal Process.* **40**(4), 882–893.

Boland, M.V. & Murphy, R.F. (2001) A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics* **17**(12), 1213–1223.

Breen, E. & Jones, R. (1996) Attribute openings, thinnings, and granulometries. *Comput. Vis. Image Understand.* **64**(3), 377–389.

Can, A., Shen, H., Turner, J.N., Tanenbaum, H.L. & Roysam, B. (1999) Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. *IEEE Trans. Inform. Technol. Biomed.* 3(2), 125–138.

Canny, J. (1983) *Finding edges and lines in images*. Technical Report AITR-720, MIT, Artificial Intelligence Laboratory, Cambridge.

Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M. & Goldbaum, M. (1989) Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Trans. Med. Imaging* **8**(3), 263–269.

Colchester, A.C.F., Ritchings, R. & Kodikara, N.D. (1990) Image segmentation using maximum gradient profiles orthogonal to edges. *Image Vis. Comput.* **8**(3), 211–217.

Eberly, D. (1996) *Ridges in Image and Data Analysis*. Kluwer Academic Publishers, Boston, MA, USA.

Fischler, M.A., Tenenbaum, J.M. & Wolf, H.C. (1981) Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Comput. Graph. Image Process.* **15**(3), 201–223.

Fleagle, S.R., Johnson, M.R., Wilbricht, C.J., *et al.* (1989) Automated analysis of coronary arterial morphology in cineangiograms: geometric and physiologic validation in humans. *IEEE Trans. Med. Imaging* **8**(4), 387–400.

Gauch, J.M. & Pizer, S.M. (1993) Multiresolution analysis of ridges and valleys in grey-scale images. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(6), 635–646.

Jacob, M. & Unser, M. (2004) Design of steerable filters for feature detection using Canny-like criteria. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(8), 1007–1019.

Kirbas, C. & Quek, F. (2004) A review of vessel extraction techniques and algorithms. *ACM Comput. Surv.* **36**(2), 81–121.

Klein, A.K., Lee, F. & Amini, A.A. (1997) Quantitative coronary angiography with deformable spline models. *IEEE Trans. Med. Imaging* **16**(5), 468–482.

Lagerstrom, R., Sun, C. & Vallotton, P. (2008) Boundary extraction of linear features using dual paths through gradient profiles. *Pattern Recogn. Lett.* **29**(12), 1753–1757.

Lang, V., Belyaev, A.G., Bogaevsici, I.A. & Kunii, T.L. (1997) Fast algorithms for ridge detection. *Proceedings of the International Conference* *on Shape Modeling and Applications*. pp. 189–197, Aizu-Wakamatsu, Japan.

Lichtenstein, N., Geiger, B. & Kam, Z. (2003) Quantitative analysis of cytoskeletal organization by digital fluorescent microscopy. *Cytometry A* **54A**(1), 8–18.

Meijering, E., Jacob, M., Sarria, J.-C.F. & Unser, M. (2003) A novel approach to neurite tracing in fluorescence microscopy images. *Proceedings of the 5th IASTED International Conference on Signal and Image Processing* (ed. by M.H. Hamza), pp. 491–495. Honolulu.

Monga, O., Armande, N. & Montesinos, P. (1997) Thin nets and crest lines: application to satellite data and medical images. *Comput. Vis. Image Understand.* **67**(3), 285–295.

Nelson, R.C. (1994) Finding line segments by stick growing. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(5), 519–523.

Neubeck, A. & Van Gool, L. (2006). Efficient non-maximum suppression. *Proceedings of International Conference on Pattern Recognition*, vol. 3. IEEE, pp. 850–855, Hong Kong.

Nevatia, R. & Babu, K.R. (1980) Linear feature extraction and description. *Comput. Graph. Image Process.* **13**, 257–269.

Ramm, P., Alexandrov, Y., Cholewinski, A., Cybuch, Y., Nadon, R. & Soltys, B.J. (2003) Automated screening of neurite outgrowth. *J. Biomol. Screen.* **8**(1), 7–18.

Serra, J. (1982) *Image Analysis and Mathematical Morphology*. Academic Press, London, UK.

Soille, P. & Talbot, H. (2001) Directional morphological filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1313–1329.

Sonka, M., Winniford, M.D. & Collins, S.M. (1995) Robust simultaneous detection of coronary borders in complex images. *IEEE Trans. Med. Imaging* **14**(1), 151–161.

Staal, J., Abramoff, M.D., Niemeijer, M., Viergever, M.A. & van Ginneken, B. (2004) Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **23**(4), 501–509.

Steger, C. (1998) An unbiased detector of curvilinear structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(2), 113–125.

Sun, C. (2001) Fast algorithm for local statistics calculation for N-dimensional images. *J. Real-Time Imaging* **7**(6), 519–527.

Sun, C. & Appleton, B. (2005) Multiple paths extraction in images using a constrained expanded trellis. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(12), 1923–1933.

Sun, C. & Pallottino, S. (2003) Circular shortest path in images. *Pattern Recogn.* **36**(3), 711–721.

Sun, C. & Vallotton, P. (2006) Fast linear feature detection using multiple directional non-maximum suppression. *Proceedings of International Conference on Pattern Recognition*, vol. 2, pp. 288–291, Hong Kong.

Sun, C., Vallotton, P., Wang, D., Lopez, J., Ng, Y. & James, D. (2009) Membrane boundary extraction using circular multiple paths. *Pattern Recogn.* **42**(4), 523–530.

Ter Haar Romeny, B.M. (2003) *Front-End Vision and Multi-Scale Image Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, The Netherlands.

Tolias, Y.A. & Panas, S.M. (1998) A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *IEEE Trans. Med. Imaging* **17**(2), 263–273.

Van de Wouwer, G., Nuydens, R., Meert, T. & Weyn, B. (2004) Automatic quantification of neurite outgrowth by means of image analysis. *Proceedings of SPIE, Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XI* (ed. by J.A. Conchello, C.J. Cogswell, & T. Wilson), vol. 5324, pp. 1–7.

Zana, F. & Klein, J.C. (2001) Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Trans. Image Process.* **10**(7), 1010–1019.

Zhou, Y.T., Venkateswar, V. & Chellappa, R. (1989) Edge detection and linear feature extraction using a 2-D random field model. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(1), 84–95.