

Fast retinal vessel tree extraction: A pixel parallel approach

C. Alonso-Montes^{1,*}, D. L. Vilarino², P. Dudek³ and M. G. Penedo¹

¹*Department of Computer Science, University of A Coruña, A Coruña, Spain*

²*Department of Electronics and Computer Science, University of Santiago de Compostela, Santiago, Spain*

³*Department of Electrical and Electronics Engineering, University of Manchester, Manchester, U.K.*

SUMMARY

Early ocular disease diagnosis is an important field in medical research. From the image processing point of view, many strategies and algorithms have been developed to deal with the extraction of the retinal vessel tree. Although reliable and accurate results have been obtained, the main disadvantage in most of these proposals is the high computation effort required. In this paper, a methodology to extract the retinal vessel tree has been developed and tested in a fine-grain pixel-parallel processor array. An analysis of the execution time has been made to demonstrate its capabilities regarding the computation speed. Moreover, an analysis of the accuracy using a publicly available database has been made to validate the algorithm performance. Copyright © 2008 John Wiley & Sons, Ltd.

Received 15 April 2008; Accepted 27 April 2008

KEY WORDS: active contours; vessel segmentation; cellular arrays; parallel processing

1. INTRODUCTION

Identification and measurement of blood vessels allow quantitative evaluation of clinical features, used for early diagnosis [1] and effective monitoring of therapies in retinopathy [2]. A lot of research has been focused on the development of algorithms and strategies for the extraction of the retinal vessel tree [3], especially with respect to the accuracy required for more precise and repeatable diagnostic methods. Among other techniques, active contours have been shown as a flexible tool for vessel extraction, mainly due to their ability to exploit the mixed control; both bottom-up (image data) and top-down (prior approximate knowledge about the location, shape and dimension of the structures) of segmentation [4]. Active contours can manage reasonably noisy, indistinguishable or ambiguous object boundaries, quite common in retinal images. However, the

*Correspondence to: C. Alonso-Montes, Department of Computer Science, University of A Coruña, A Coruña, Spain.

†E-mail: calonso@udc.es

Contract/grant sponsor: Xunta de Galicia; contract/grant number: PGIDIT06TIC10502PR

main disadvantage of this kind of techniques is the high computation effort. Parallel hardware-based approach can offer a solution providing high computation speed that is required.

Some retinal vessel processing techniques have been proposed [5, 6] intended to improve the execution time by means of developing all the operations under the cellular neural network (CNN) [7] paradigm; this approach assumes efficient processing in a pixel-parallel way. However, some of the proposed CNN-based operations utilize non-linear templates, which prevent their implementation in the current generation of cellular processor VLSI chips.

In this paper, the original proposal addressed in [5] has been redefined in terms of local dynamic convolutions and morphological operations together with arithmetic and logical operations to be implemented and tested in a fine-grain single instruction multiple data (SIMD) parallel processor array. This methodology finds the exterior of the vessels using an active contour technique, the so-called pixel-level snakes (PLS) [8]. PLS have been previously implemented on hardware architectures with capabilities of SIMD processing, such as the CNN-based ACE4K chip [9] as well as the focal plane cellular processor array SCAMP-3 vision system [10, 11]. The performance and the accuracy of the proposed method have been tested using the publicly available digital retinal images for vessel extraction (DRIVE) database [12].

This paper is structured as follows: in Section 2 the PLS performance is briefly described, in Section 3 the proposed method is explained, in Section 4 the main results are shown, and, finally in Section 5 the main conclusions are presented.

2. PIXEL-LEVEL SNAKES

PLS are a massively parallel active contour technique inspired by the energy-based deformable models. The inputs consist of a binary image containing the initial contours and a multi-bit image, called external potential, which defines the desired segmentation boundaries to guide the contour evolution.

PLS operate along the four cardinal directions performing the evolution as a pixel-by-pixel shift (activation and deactivation of pixels in a binary image). The goal after each cycle (four iterations, one for each cardinal direction) is to obtain a new contour slightly shifted in order to be closer to the boundaries of interest. Like in conventional active contours, the evolution is controlled by means of the internal, the external, and the balloon potentials. The internal potential controls the smoothing effect of the contour giving more robustness to the model against noise. The external potential guides PLS towards the boundaries of interest, which should coincide with the valleys of a potential field. The balloon potential controls the inflation or deflation tendency of a closed contour. Topological changes, such as merging and splitting contours, are also easily handled by PLS, as well, as preserving the topology when it is needed. The inputs consist of a binary image containing the initial contours and a multi-bit image containing the guiding information image, called external potential, which guides the PLS evolution. A comprehensive description of PLS can be found in [8].

Since PLS were introduced, several algorithms implementing this cellular active contour technique have been proposed to optimize the computation performance execution on fine-grain pixel-parallel processor arrays [13]. In this paper, the PLS region-based approach proposed in [11] has been used, where the contours are implicitly represented as the boundaries of active regions. This algorithm proved to be particularly efficient in applications where the contour evolution is always expansive or compressive, specially regarding the computation speed.

3. ALGORITHM

The proposed algorithm has been designed in order to automatically extract the vessels, computing the main input images needed by the PLS (the initial region and the external potential images) based on local statistics of the images under processing. The main stages proposed in the algorithm are shown in Figure 1. In the first stage, some pre-filtering operations are performed to improve the signal-to-noise ratio in order to pre-estimate the regions between the vessels. Then the initial contours and the external potential images are computed in an automatic way. Using both images, PLS will evolve to fit the vessel edges.

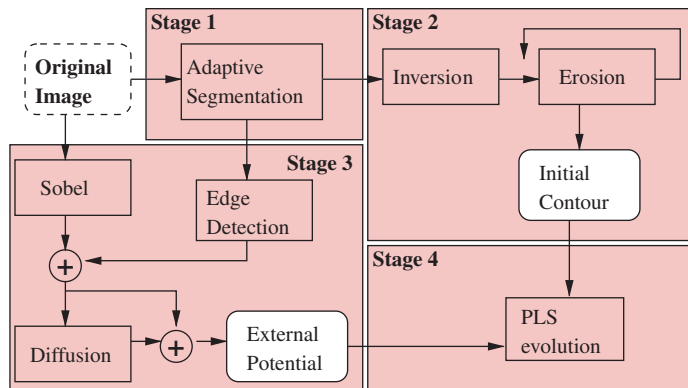


Figure 1. Flow diagram with the building blocks of the proposed algorithm: *Stage 1*, vessel region pre-estimation; *Stage 2*, initial region estimation; *Stage 3*, external potential estimation; *Stage 4*, PLS evolution.

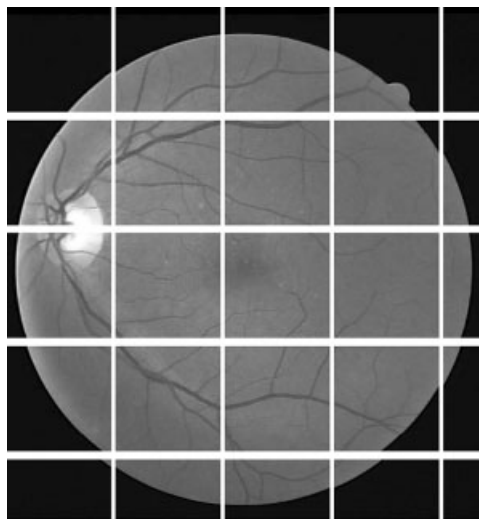


Figure 2. Original angiography split into 128×128 subwindows.

Note that fitting the interior of the vessels has been the most common technique used [14, 15]. However, this strategy has some disadvantages. The internal potential depends on the local curvature of the contour. In those cases where the contours flow inside tubular structures (such as the vessels), the internal potential has a high value, stopping the contour evolution due to its strength with respect to the external potential. Therefore, the evolutions is controlled only by means of the external and balloon potentials, and complex rules should be defined to avoid the contour flowing outside the vessels. Moreover, the initialization process is more complex, since fewer pixels belong to the vessels than the background, e.g. in the images of the DRIVE database, 12.7% of pixels belong to the vessel [16]. In this paper, fitting the exterior of the vessels has been proposed in order to avoid all those disadvantages, providing more robustness and control over the PLS evolution.

The proposed algorithm has been developed to be computed in a hardware platform with a pixel processor array. Owing to the high resolution of the retinal images, the original image is proposed to be split into $N \times N$ subwindows in order to fit the size of the current processor array integrated circuit implementations without losing the image resolution information (see Figure 2).

In the following sections, the main stages of the proposed algorithm are explained.

3.1. Stage 1: Vessel region pre-estimation

The main goal of this stage is to pre-estimate the vessel edges. Due to the non-uniformity of the grey-level values along the vessels, an adaptive segmentation is needed. Instead of the complex CNN-based approach addressed in [5], another strategy suitable for a hardware implementation has been employed (Figure 3).

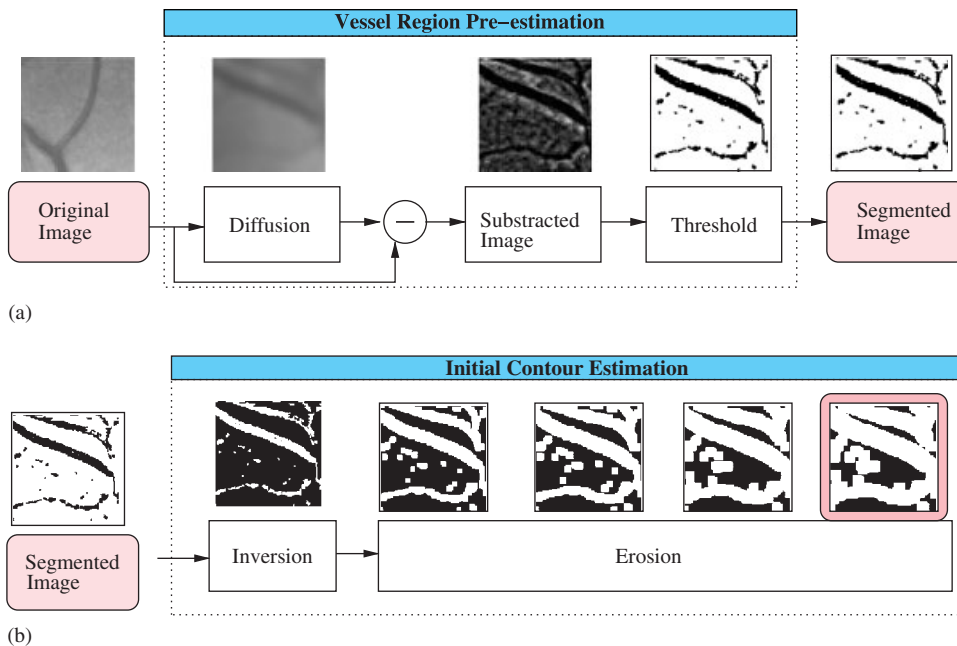


Figure 3. (a) Stage 1: Vessel region pre-estimation and (b) Stage 2: Initial region estimation.

Firstly, the original image is blurred by means of a diffusion step. This blurred image is used to determine the local threshold value that properly segments not only vessels with a high contrast but also weak vessels. The blurred image is subtracted from the original one, and finally, the result is binarized based on a fixed threshold value.

3.2. Stage 2: Initial region estimation

The aim of this stage is to obtain a suitable initialization of the contour locations (Figure 3(b)). An inversion of the segmented image is initially made to define the regions, represented by black pixels, between the vessels. Since this image can contain vessel discontinuities and regions inside the vessels, it is eroded several times. Thus, the obtained image will contain only regions situated completely outside of the vessel locations. The output of the erosion step will represent the initial conditions for PLS.

3.3. Stage 3: External potential estimation

The aim of this stage is to compute a suitable external potential image to guide the PLS evolution. The external potential image determines a potential field that guides PLS towards the exterior of the vessel edges. The processing steps performed in this stage are illustrated in Figure 4.

Firstly, the Sobel operator is applied to the original image to get the actual vessel edges. Although this operator does not introduce much noise, vessel discontinuities appear, and low contrast vessels are not properly segmented. On the other hand, the image obtained in Stage 1 contains clearly defined vessel edges but at the cost of more segmented noise and inaccuracy in the vessel locations. Therefore, a combination of both images is used in order to properly guide the PLS evolution. Then the result is diffused to obtain a suitable potential field gradient from PLS to flow through. PLS will evolve guided by a stronger external potential in areas close to the edge due to a higher potential value, and *vice versa*. Finally, since a diffusion homogenizes the grey-level values, especially in the edge locations, the weighted edge image is added again to emphasize the vessel edges.

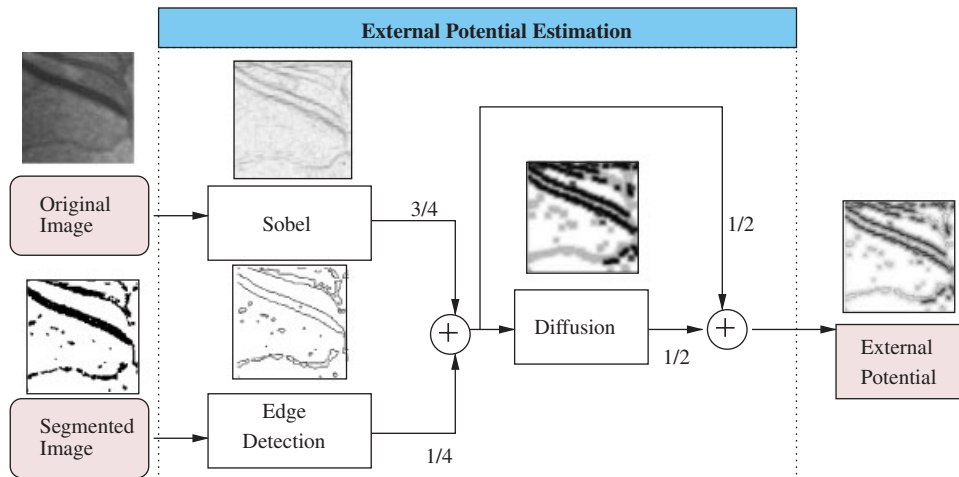


Figure 4. Stage 3: External potential estimation.

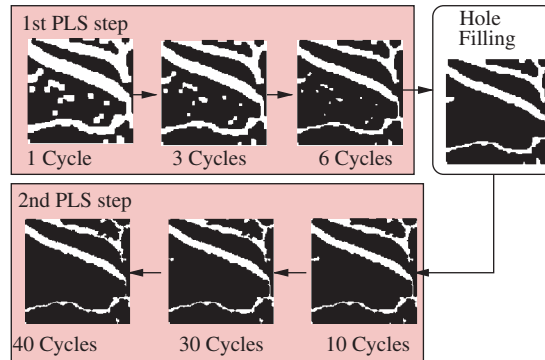


Figure 5. Stage 4: PLS evolution.

3.4. Stage 4: PLS evolution

PLS evolve to fit the exterior of the vessels using both the initial region and the external potential images, previously computed. The main goal of this stage is to determine the parameters that control the evolution. The external potential guides the PLS evolution towards the vessel edges, whereas the internal potential is used to avoid PLS evolving through vessel discontinuities. Since the vessel edges are situated outside of the initial regions, an inflation potential can help to move the contours in those cases where the external potential is too weak. Taking into account all these considerations, this stage has been split into several steps (see Figure 5): a first PLS step, then a hole filling operation, and finally a second PLS step.

During the first PLS step, the evolution is mainly controlled by means of the balloon potential. The external potential is too weak to move the regions, because these are located far from the vessel edges. Therefore, the movement of the regions is mainly controlled by means of the strength of the balloon potential. Region merging is allowed during this stage, since a large region could be split into smaller ones after the erosion operation during the initial region estimation. Moreover, due to the distance to the vessel edges, the region merging will not affect the vessel topology. During this step, the internal potential will provide an uniform evolution of the regions. Then, a hole filling operation is used to remove internal regions that have appeared due to noise, segmented during the previous stages (see Figure 5). Finally, during the second PLS step, PLS fit the vessel edges relying mainly on the external potential due to the proximity to the vessel edge locations. In this case, the influence of the balloon potential is weaker than the external potential, which is strong enough to guide the region evolution. In this step, region merging is prevented in order to maintain the vessel topology. Furthermore, the internal potential has a higher influence, since it prevents the evolution through small cavities or discontinuities in the vessel topology.

4. RESULTS

The complete set of 40 retinal images provided by the DRIVE database has been used to test the performance, accuracy, and the execution time required for the proposed algorithm. The retinal images (including seven with pathologies) have been captured by a Canon CR5 3CCD camera with

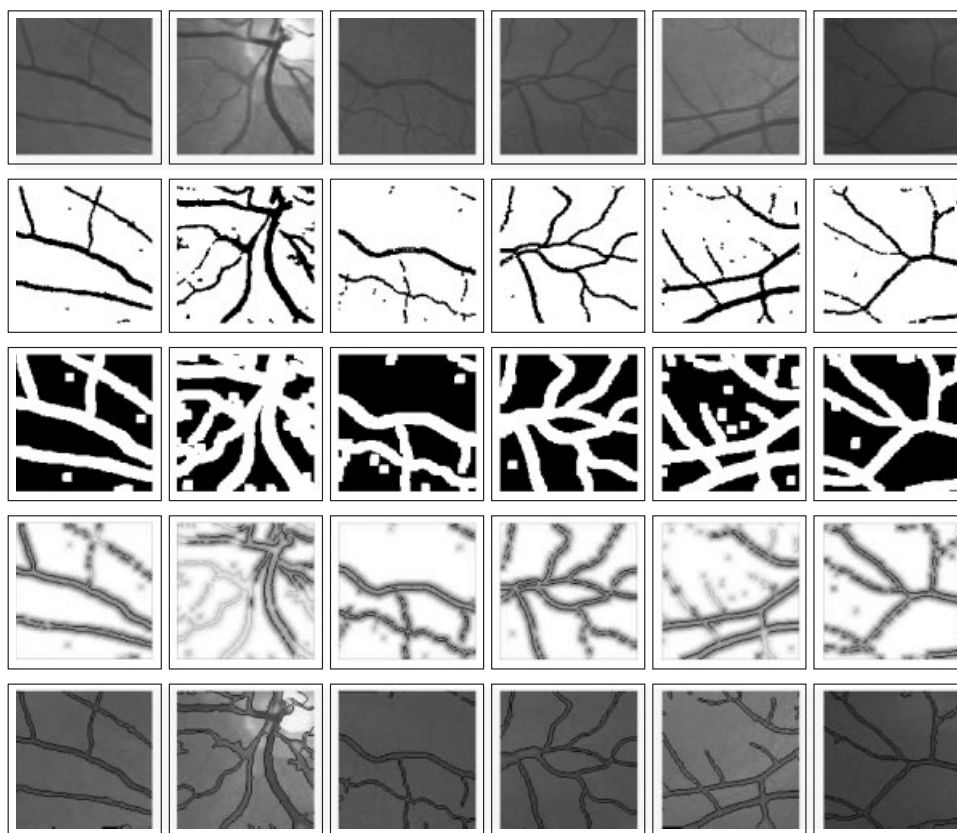


Figure 6. Output images obtained in each stage: *First row*, original; *second row*, pre-estimated vessel regions; *third row*, initial regions; *fourth row*, external potential images; *fifth row*, PLS contours superimposed over the original images.

a 45° field of view (FOV). Each of the images has been split into 128×128 subwindows in order to fit the size of the current processor array, particularly on the SCAMP-3 vision system [13].

The result obtained in each of the proposed stages are shown in Figure 6.

Stage 1: During this stage, the vessel edges are pre-estimated. The adaptive segmented results are shown in the second row of Figure 6. Note that low contrast vessels are properly segmented, since the adaptive segmentation is based on local statistics of the image.

Stage 2: The segmented image from the first stage is inverted in order to define the regions between the vessels. These regions are eroded four times to ensure that all of them are completely outside of the vessels (third row of Figure 6).

Stage 3: Through the combination of the Sobel result and the image from Stage 1, vessel continuity information is maintained in low contrast vessels (fourth row of Figure 6).

Stage 4: From the initial regions (third row) and using the external potential images (fourth row), PLS evolve to fit the actual vessel edges. During the first PLS step, a high value potential is used during six cycles. This number of cycles has been selected based on the number of erosion steps used during Stage 2, which gives an approximation of the distance to the vessel edges. Since

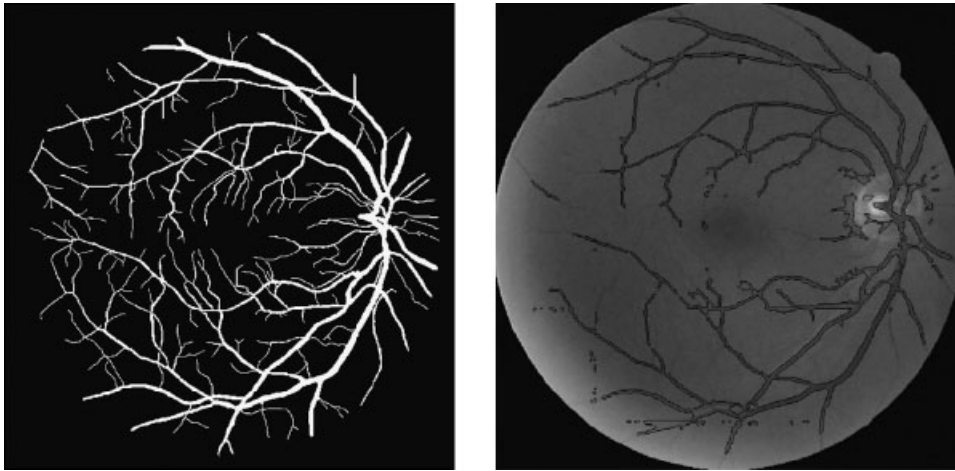


Figure 7. *Left*: ground truth from the DRIVE database and *right*: obtained retinal vessel tree (PLS final contour result).

the adaptive segmentation gives only a pre-estimation of the vessel edges, the number of cycles used during the first PLS step could be increased. Then, the hole filling is applied to remove noisy regions, and finally, the second PLS step is performed. The number of cycles during this last step has been empirically established at 40 cycles, since this number of cycles is sufficient in the PLS evolution in all the subwindows. Note the adjustment of PLS to the vessel edges (last row of Figure 6). The whole retinal vessel tree obtained by means of the methodology can be seen in Figure 7, which consists of the union of all the processed subwindows.

The criterion used to evaluate the performance of the proposed algorithm is the average value of the accuracy for all the images (maximum average accuracy, MAA). The accuracy of each image has been computed according to the formulation proposed in [16]:

$$\text{Accuracy} = \frac{T_{\text{pos}} + T_{\text{neg}}}{\text{NP}} \quad (1)$$

where T_{pos} and T_{neg} are the vessel (true positive) and non-vessel (true negative) pixels correctly classified, respectively, and NP is the number of pixels considered in the FOV region. Two sets of manually segmented images made by experts (called first and second observer) are provided in the DRIVE database. The images, segmented by the second specialist, have been used as ground truth to compute the number of pixels correctly classified (MAA) with our methodology. The MAA of our algorithm can be compared with other PC-based algorithms found in the literature, see Table I. Furthermore, a comparison between the two manual segmentations has been made in [16] in order to show the difficulty of the segmentation task, even for specialists. Although there are proposals that can achieve a similar MAA, the required computation time is significantly higher than with our algorithm.

Finally, a test of the computation speed on an actual parallel processor, the SCAMP-3 vision system, has been analysed. The SCAMP-3 vision system executes a sequence of simple array instructions, such as addition, inversion, one-neighbour access, operating in a pixel-parallel manner on 128×128 arrays, at a rate of 1.25 MOPS per pixel. The execution time required to perform

Table I. MAA and the execution time.

	Manual method	Soares <i>et al.</i> [17]	Chaudhuri <i>et al.</i> [18]	Our proposal
MAA	0.9473	0.9466	0.8773	0.9180
Execution time	2 h	3 min	50 s	0.1925 s

Table II. SCAMP execution time for each stage, in a 128×128 subwindow.

No. stage	Stage	Execution time (μ s)
1	Vessel region pre-estimation	12.8
2	Initial region estimation	55.2
3	External potential estimation	134.4
4	First PLS step	518
	Hole filling	1954.5
	Second PLS step	3870.8

the whole algorithm for a 128×128 subwindow in the SCAMP-3 vision system is about 6.5 ms (see the execution time required for each stage in Table II). The total I/O time required for a 128×128 image is 1.25 ms [19]. Since the standard retinal image size is 768×584 , a total number of 30 subwindows is required. Therefore, the global execution time required to process the whole retinal vessel tree is about 0.23 s. The execution time can be compared with the PC-based software implementations shown in Table I.

An overlapping technique has been studied in order to improve the MAA value, since vessel continuity can be lost in the limits of the subwindows. This technique consists of overlapping rows and columns in order to obtain redundant information in the limit areas of the subwindow. Then, the union of all the subwindows is made taking into account the overlapping area. Four main criteria have been studied in order to see the influence of the overlapping in the final result: accuracy, specificity, sensitivity, and execution time. Specificity and sensitivity are the true negative and the true positive ratios, respectively, expressed as percentages. According to our analysis, although the sensitivity value is improved with the overlapping size, specificity and accuracy have a slight improvement only (see Figure 8), and on the other hand, the required execution time increases significantly due to the increment of the subwindow number which should be processed (see Table III).

5. CONCLUSION

In this paper we present an algorithm for the retinal vessel extraction optimized for its implementation on a cellular processor array. The algorithm is based on the CNN-based methodology proposed in [5], which could not be implemented in a current generation of CNN chips due to the use of non-linear templates for some of the proposed steps. In this paper, a redefined approach has been proposed in order to be implemented in a SIMD parallel cellular processor array architecture. From the image processing point of view, the accuracy obtained is suitable according to the test made against the DRIVE database. Moreover, the execution time obtained using the SCAMP-3

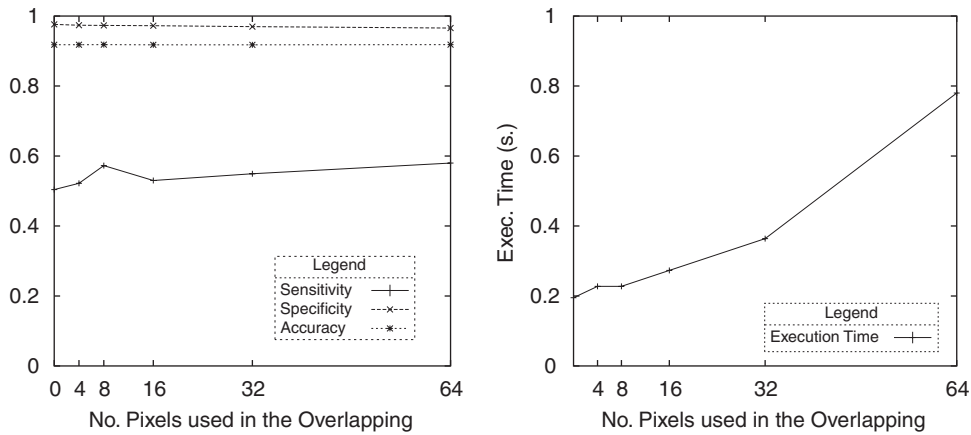


Figure 8. Overlapping size influence in specificity, sensitivity, MAA, and execution time parameters.

Table III. Execution time and MAA depending on the overlapping size.

Overlapping size	Number of subwindows	MAA	Global execution time (s)
0	30	0.9180	0.192
4	35	0.9181	0.227
8	35	0.9183	0.227
16	42	0.9178	0.273
32	56	0.9177	0.364
64	120	0.9182	0.780

system is much shorter (several orders of magnitude) than conventional PC-based applications. Our algorithm is not only suitable for being implemented in an actual parallel processor but also the accuracy value is adequate to be taken into account for medical applications. The obtained retinal vessel tree can be used for a wide range of applications, such as vessel measurement, ocular disease diagnosis applications and even for vessel pattern-based authentication systems.

ACKNOWLEDGEMENTS

The authors would like to thank Xunta de Galicia, which has partly supported this work (PGIDIT06TIC10502PR).

REFERENCES

- Brieva J, Galvez M, Toumoulin C. Coronary extraction and stenosis quantification in X-ray angiographic imaging. *IEEE-EMBC Proceedings of the International Conference on Engineering in Medicine and Biology Society*, Mexico City, Mexico, vol. 1, 2004; 1714–1717.
- Miles FP, Nuttall AL. Matched filter estimation of serial blood vessel diameters from video images. *IEEE Transactions on Medical Imaging* 1993; **12**:147–152.
- Kirbas C, Quek F. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys* 2004; **36**(2):81–121.

4. McInerney T, Terzopoulos D. Deformable models in medical image analysis: a survey. *Medical Image Analysis* 1996; **1**:91–108.
5. Alonso-Montes C, Vilarino DL, Penedo MG. CNN-based automatic retinal vascular tree extraction. *IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA)*, Hsin-Chu, Taiwan, 2005; 61–64.
6. Perfetti R, Ricci E, Casali D, Costantini G. Cellular neural networks with virtual template expansion for retinal vessel segmentation. *IEEE Transactions on Circuits and Systems II* 2007; **54**:141–145.
7. Chua LO, Roska T. The CNN paradigm. *IEEE Transactions on Circuits and Systems I* 1993; **40**:147–156.
8. Vilariño DL, Rekeczky C. Pixel-level snakes on the CNUM: algorithm design, on-chip implementation and applications. *International Journal of Circuit Theory and Applications* 2005; **33**:17–51.
9. Vilariño DL, Rekeczky C. Implementation of a pixel-level snake algorithm on a CNUM-based chip set architecture. *IEEE Transactions on Circuits and Systems I* 2004; **51**:885–891.
10. Dudek P, Carey SJ. General-purpose 128×128 SIMD processor array with integrated image sensor. *Electronics Letters* 2006; **42**(12):678–679.
11. Dudek P, Vilariño DL. A cellular active contours algorithm based on region evolution. *IEEE International Workshop on Cellular Neural Networks and their Applications*, Istanbul, Turkey, 2006; 269–274.
12. Drive. *Drive: Digital Retinal Images for Vessel Extraction*, 2007 (available from: <http://www.isi.uu.nl/Research/Databases/DRIVE/>).
13. Vilariño DL, Dudek P. Evolution of pixel level snakes towards an efficient hardware implementation. *IEEE International Symposium on Circuits and Systems (ISCAS)*, New Orleans, U.S.A., 2007.
14. Caderno IG, Penedo MG, Mariño C, Carreira MJ, Gomez-Ulla F, González F. Automatic extraction of the retina AV index. *International Conference on Image Analysis and Recognition (ICIAR'04)*. Lecture Notes in Computer Science, vol. 3212. Springer: Berlin, 2004; 132–140.
15. McInerney T, Terzopoulos D. Topologically adaptable snakes. *Proceedings of the 5th IEEE International Conference on Computer Vision*, Los Alamitos, CA, U.S.A., 1995; 840–845.
16. Niemeijer M, Staal J, van Ginneken B, Loog M, Abramoff M. Comparative study of retinal vessel segmentation methods on a new publicly available database. In *SPIE Medical Imaging*, vol. 5370, Fitzpatrick JM, Sonka M (eds). SPIE: San Diego, CA, U.S.A., 2004; 648–656.
17. Soares JVB, Leandro JGG, Cesar RM, Jelinek HF, Cree MJ. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging* 2006; **25**:1214–1222.
18. Chaudhuri S, Chatterjee S, Katz N, Nelson M, Goldbaum M. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging* 1989; **8**:263–269.
19. Barr DRW, Carey SJ, Lopich A, Dudek P. A control system for a cellular processor array. *IEEE International Workshop Cellular Neural Networks and their Applications*, Istanbul, Turkey, 2006; 176–181.