Facultade de Informática
Departamento de Computación

PhD Thesis

Automatic Pixel-Parallel Extraction of the Retinal Vascular Tree: Algorithm Design, On-Chip Implementation and Applications

Author: Carmen Alonso Montes
Supervisors: Manuel F. González Penedo
David López Vilariño

D. Manuel Fco. González Penedo, Profesor Titular de Universidad do Departamento de Computación da Universidade da Coruña.

y

D. David López Vilariño, Profesor Contratado Doutor do Departamento de Electrónica y Computación da Universidade de Santiago de Compostela

CERTIFICAN que:

La memoria que se presenta, titulada *Automatic Pixel-Parallel Extraction of the Retinal Vascular Tree: Algorithm Design, On-Chip Implementation and Applications* ha sido realizada por Dña. Carmen Alonso Montes bajo nuestra dirección en el Departamento de Computación da Universidade da Coruña y constituye la tesis que presenta para optar al grado de Doctor.

Y para que así conste firmamos la presente en

A Coruña, a 2 de Mayo de 2008.

Fdo.: Manuel Fco. González Penedo   Fdo. David López Vilariño

Acknowledgements

I would like to thank to my supervisors, Manuel Fco. González Penedo and David López Vilariño, for all their help and wise advice throughout this thesis. Specially, I am very grateful for their patient an enthusiasm with this work and the results obtained.

I would also like to express my sincere gratitude to the University of A Coruña, and particularly to the Department of Computer Science and to the Department of Electronics and Computer Science, from the University of Santiago de Compostela, for their support in the development of this research.

I strongly thank for the valuable advice and wise guidance of Dr. Piotr Dudek, form the School of Electrical and Electronic Engineering during my research stays at the University of Manchester. As a special acknowledgment, I want to thank to all the members of the laboratory directed by Dr. Piotr Dudek, specially to Henry Kite, David Barr and Christian Brox-Nielsen for making me easier my stays in Manchester, and also to Alex Lopich, Jaywan Wijekoon and Steve Carey for their support and patient.

I want to thank present and past members of the staff from VARPA laboratory: Noelia, Marianne, Marta, Cas, Marcos, José and María for for their support and the help they gave me through all my PhD. Particularly, I would like to thank Marcos Ortega for his valuable collaboration in the development of this work.

Finally, I would like to mention my family and friends, for their help and the great interest showed during the elaboration of my PhD thesis. Anita, Fran, Andrés, Cris, Dani, Sandra, David, María ... thanks for your help and for believing in me.

To my sister Anita, my mother Consuelo and my father Servando; without their love I could not finish this thesis. Danke schön.

"Time is a great teacher, but unfortunately it kills all its pupils"

Hector Berlioz

Preface

Currently, medical images have become an important field of research due to the advances in their acquisition, storage and management in a wide range of applications. Regarding the medical domain, a great effort has been devoted to develop tools and applications related with disease diagnosis, bone reconstruction, identification of anatomical structures, etc. Particularly, the extraction of the vessel pattern from retinal images plays an important role not only in the medical tools; but also in biometric identification applications. A lot of research has been devoted to accurately extract the retinal vessel tree in order to define automated applications for early ocular disease diagnosis or vessel-pattern-based authentication applications.

From the image processing point of view, the special features of retinal images, such as noise, low contrast vessel or high gray-level variability along the vessel structures make complex and difficult the processing of this kind of images. Although a lot of techniques and algorithms obtain promising results, the computation effort required for doing those task, is still a problem for most of them.

In this thesis, an algorithm for the automatic extraction of the retinal vessel tree is developed, specially defined in terms of fine grain SIMD (Single Instruction Multiple Data) processing with the purpose of improving the computation time. This algorithm is based on a pixel-parallel active contour based technique, the so-called Pixel Level Snakes (PLS). PLS will evolve to fit the exterior of the vessels. A custom implementation has been made to optimise the computation speed, and the calibration of the PLS parameters has been estimated for the specific retinal vessel tree extraction. This pixel parallel algorithm has been tested from the image processing point of view, using the images available in the Digital Retinal Images for Vessel Extraction database (DRIVE). The accuracy could be compared with other proposals found in the bibliography. The time performance analysis has been made using the SIMD chip. Both features, the reliability and the time performance, show the capability of the proposed algorithm. In this sense, this algorithm maintains a good accuracy level as well as the execution time is its great advantage, since it is one or several orders of magnitude less than conventional PC-based algorithms or techniques.

The retinal vessel tree extracted using the algorithm proposed in this thesis has been also integrated into practical applications, an authentication and a medical applications. The analysis of the performance of all those applications, shows that the retinal vessel tree obtained by the algorithm is suitable for them, obtaining similar final values.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Currently, the advances on digital imaging systems and the storage of digital images offer very high-resolution images that can be used in many applications from several areas like medicine or surveillance. Digital imaging is easier to storage preserving its quality with the time. The transmission of this kind of images makes them suitable to integrate their use into large scalable systems. Moreover, these images can be improved and even be subjected to techniques to perform an objective quantitative and qualitative analysis. As for the medical image processing is concerned, a lot of research has been focused on the computation of accurate geometric models of anatomic structures from medical images in order to exploit the potential of automated applications for early disease diagnosis, security tasks, model bone reconstruction, and so on.

Retinal vessel features plays an important role in medical research for early diagnosis [Patton et al., 2006], such as diabetes [Jelinek et al., 2007, Pedersen et al., 2000], or cardiovascular risk [Brieva et al., 2004, Gao et al., 2001b, Couper et al., 2002]; and effective monitoring of therapies in retinopathy [Gomez-Ulla et al., 2002, Wong et al., 2001, Hubbard and Brothers, 1999, Pose-Reino et al., 2005, Zhou et al., 1994, Wong et al., 2002]; as well as vessel pattern-based authentication applications [Mariño et al., 2006, Ortega et al., 2006b], since the unique structure of the blood vessels makes it suitable to be used for biometric identification [Bolle et al., 1998, Uludag et al., 2004]. In the medical domain, the analysis of quantitative retinal vessel features is the basis of research tools [Mosquera et al., 2003, Li et al., 2005, Caderno et al., 2004] intended to establish the relationships between the retinal microvasculature and cardiovascular disease, diabetes diagnosis, etc. These diagnostic systems can be used in large-scales screening programs, with the potential for significant resource savings. A lot of effort has been devoted to the development of techniques and algorithms that compute efficiently and accurately retinal vessel tree features from medical images, particularly using fundus retinal images. Although promising techniques have been proposed, the improvement of the computation effort is still the main issue. In this sense, hardware based approaches can speed up those time performance leading towards the integration into real-time applications.

In this chapter an introduction to the main issues and problems of computing the retinal vessel tree is presented. First, a brief introduction about the different types of medical images and retinal cameras is made. Then, the state of the art in the retinal vessel tree extraction is presented. Finally, the retinal vessel tree extraction algorithm proposed in this thesis is discussed.

1.1 Medical Images

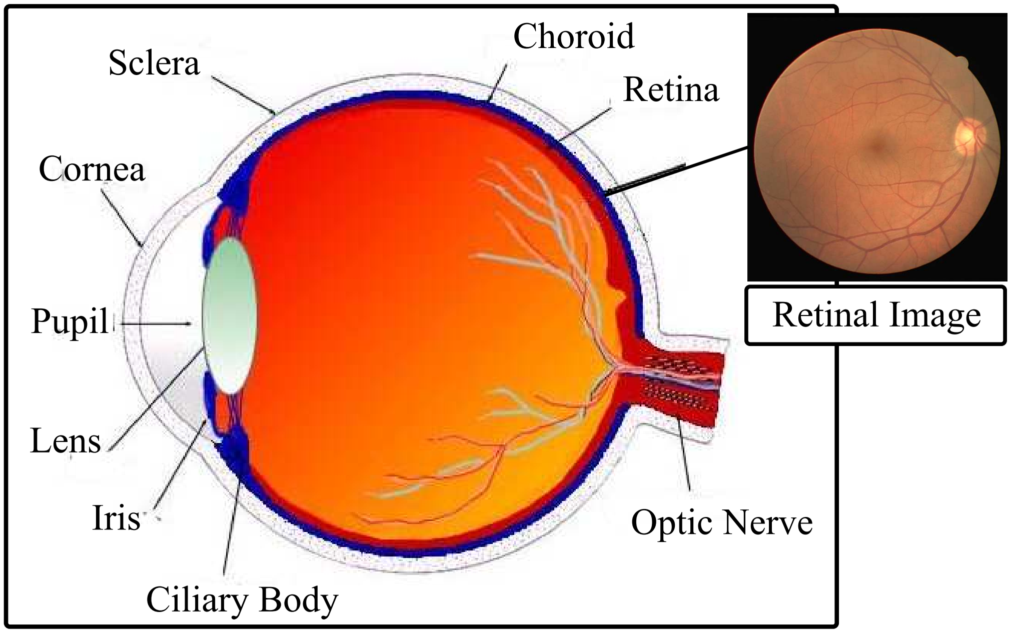

Fig. 1.1: Human eye cross-sectional view with an example of a retinal image

The **retina** is the light sensitive inner layer of the eye, which receives images formed by the lens and transmits them through the optic nerve to the brain. A retinal fundus image (see Fig. 1.1) can be defined as the interior lining of the eyeball, including the retina, optic disk, the retinal vessel tree and the macula (the small spot in the retina where vision is keenest). The fundus is the portion of the inner eye that can be seen during an eye examination by looking through the pupil. Fundus retinal images are used to study the main vessel features and the vascular structure in many applications and tools.

There are a wide range of retinal cameras dedicated to get retinal images, according to the several necessities in the medical research. Particularly, for obtaining the retinal fundus images, the cameras are classified into mydriatic and non-mydriatic retinal image cameras. The *mydriatic camera* needs the administration of dilatation drops to the patient in order to dilate the retina. This camera is used specially in those cases where the pupil of the patient is $\leq$ 4 mm. A *non-mydriatic camera* is the most common for obtaining the retinal fundus images. Some examples of well-known non-mydriatic cameras are the Non-Mydriatic Canon CR6-45NM (Canon USA, Lake Success, EEUU) and Topcon TRC-NW6S (Topcon America Corp., Paramus, EEUU), see Fig. 1.2. These cameras enable ophthalmologists and optometrists to instantly capture ultra high-resolution digital images.

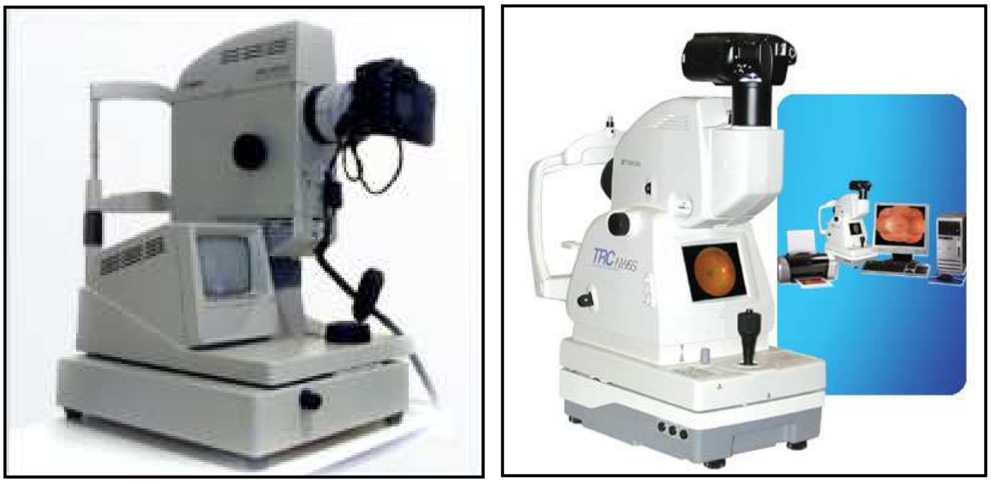

Fig. 1.2: Two types of non-mydriatic retinal image cameras: *(Left)* Canon CR6-45NM and *(Right)* Topcon TRC-NW6S

Fundus retinal images can be roughly classified into the following types (see Fig. 1.3):

- Digital fluorescein angiography
- Digital colour fundus photography

- Digital red free photography

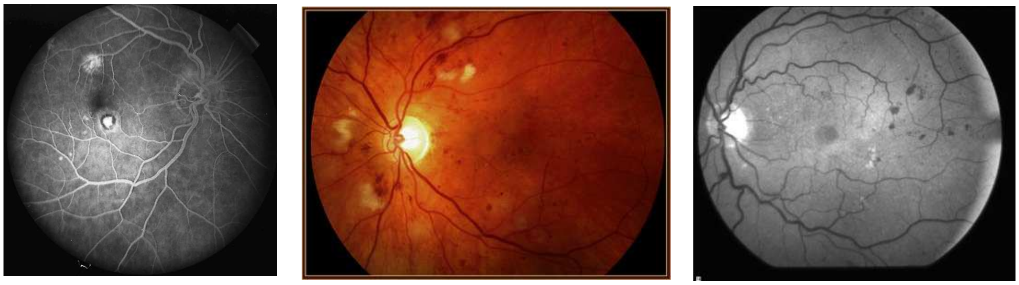

Fig. 1.3: Types of digital retinal images : *(from left to right)* digital fluorescing angiography, colour fundus photography and red free photography

The **digital fluorescing angiography** is an image obtained using the dye tracing method. Sodium fluorescein is injected into the systemic circulation, and then an angiography is obtained by photographing the fluorescence emitted after illumination of the retina with blue light at a wavelength of 490 nanometers. The fluorescein dye also reappears 12-24 hours in the patient urine, causing a yellow-green appearance. The ophthalmologist evaluates the dye patterns to find pathologic changes, such as staining, abnormal vessels, diabetic retinopathy or even tumors. However, the use of the dye has a high risk of adverse effects, and in fact it is the most intrusive technique for the patient.

The **digital colour fundus photography** is obtained using a customised camera which is mounted to a microscope with intricate lenses and mirrors. The high-powered lenses are designed in such a way that the ophthalmologist visualises the back of the eye by focusing light through the cornea, pupil, and lens. Fundus photography is used to evaluate the health of the optic nerve, vitreous, macula, retina, and its blood vessels (see Fig. 1.4).

The **digital red free photography** uses invisible infrared light to illuminate the retina during alignment and focus. So, the patient does not experience blinding white light during this process. Images are captured using a mild white zenon flash.

The adverse effects as well as the intrusiveness of the way of taking the fundus image are issues that should be taken into account depending on the application or tool. In this thesis, digital colour fundus and red free images (also called retinal images) have been selected because the method to get them is the least intrusive.

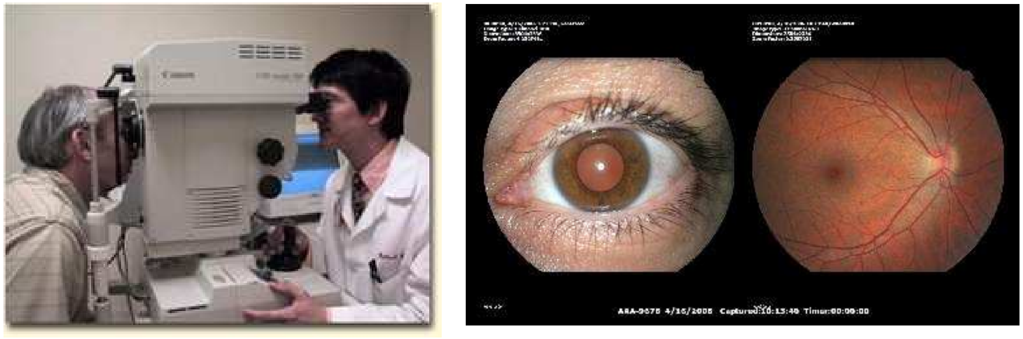

Fig. 1.4: *(Left)* CCD retinal camera and *(Right)* example of a retinal image taken from the eye

Since many processing and measurement tools are implemented to operate on gray scale, colour images must be transformed. Colour images use three channels to define the pixel intensities: *red, green and blue* (RGB), to produce an overall composite. RGB images require a three-dimensional array to convey the extra colour information. The use of only the green channel in fundus image analysis is quite common, since the contrast between the background and the vessel features is larger than in the other channels.

1.2 Retinal Vessel Tree Extraction: state of the art

Currently, due to the huge volume of patient information, particularly with medical images, automated image analysis techniques play a central role in order to avoid or alleviate manual analysis. The reliability [Couper et al., 2002] of retinal vessel tree extraction is the key in many applications and tools, specially regarding automated radiological diagnostic systems. A lot of techniques and tools have been published in the research literature to cover different aspects, features and applications related with retinal vessels, specially in the medical domain, such as optic disk detection [Trucco and Kamat, 2004, Huiqi and Chutatape, 2001, Xu et al., 2006, Foracchia et al., 2004, Hoover and Goldbaum, 2003], image registration [N. Ryan and de Chazal, 2004, Zana and Klein, 1999, Matsopoulos et al., 2004], change detection [Goldbaum et al., 1996, Hubbard and Brothers, 1999, Wong et al., 2002, Heneghan et al., 2002], pathology detection and quantification [Martinez-Perez et al., 2002], tracking in video sequences [Miles and Nuttall, 1993, Koozekanani et al., 2003], and computer-aided screening systems [Patton et al., 2006, Walter et al., 2002]. An exhaustive survey of the different methodologies, techniques and proposals is shown in [Kirbas and Quek, 2004].

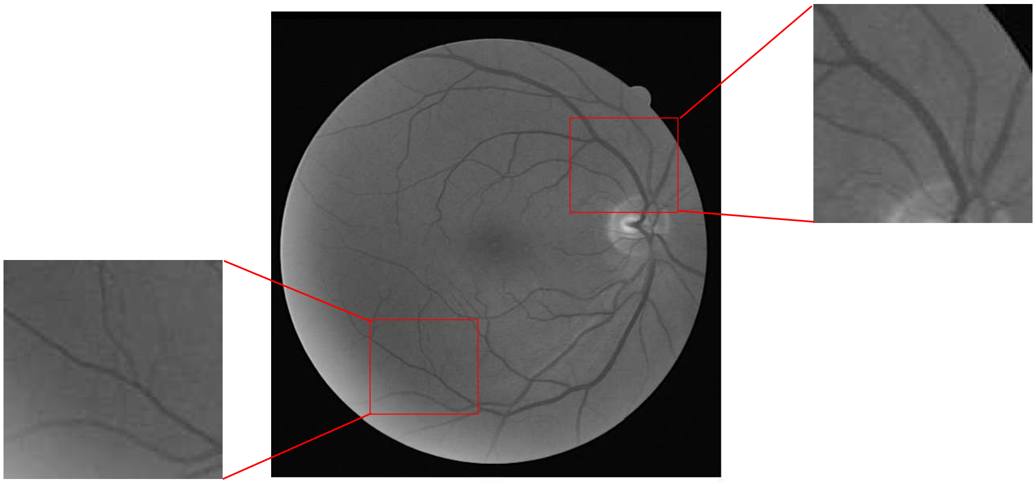

Fig. 1.5: Original retinal image and zoom of two areas of the image. Notice the difference in the gray level contrast of the vessels with respect to the background

The methods for the retinal vessel extraction can be categorised into the following groups:

- Pattern recognition techniques, which include techniques so well-known like matched filters [Al-Rawi and Karajeh, 2007, Al-Rawi et al., 2007, Sofka and Stewart, 2006, Chaudhuri et al., 1989, Hoover et al., 2000], adaptive threshold [Chanwimaluang and Fan, 2003, Jiang and Mojon, 2003], intensity edges [Koozekanani et al., 2001, Can et al., 2002] or region-based approaches [Martinez-Perez et al., 1999, O'Brien and Ezquerra, 1994, Schmitt et al., 2002]

- Model based approaches, which include classical deformable [McInerney and Terzopoulos, 1995, 1996a], or geometric deformable models [Osher and Sethian, 1988, Caselles et al., 1997]

- Tracking-based approaches [Tolias and Panas, 1998]

- Artificial intelligence-based approaches [Goldbaum et al., 1996]

- Neural networks approaches [Nekovei and Sun, 1995]

- Tube-like object detection approaches [Kompatsiaris et al., 2000]

From the algorithmic point of view, special features of retinal images will determine the performance, reliability and accuracy of the different methods for vessel extraction that can be found in the literature. The features can be roughly outlined as follows:

- There is a wide range of vessel widths, from less than a pixel up to 12 pixels wide (see Fig. 1.5)
- Vessels are usually low contrast, particularly narrow vessels (see Fig. 1.5)
- A variety of structures, such as the retina boundary, the optic disc, and pathologies can produce stronger responses at their boundaries (see Fig. 1.6)
- There is also a bright strip running down the centre of some vessels, called the *central reflex* (see Fig. 1.7), causing a complicated intensity cross-section, which makes it harder to distinguish from two side-by-side vessels

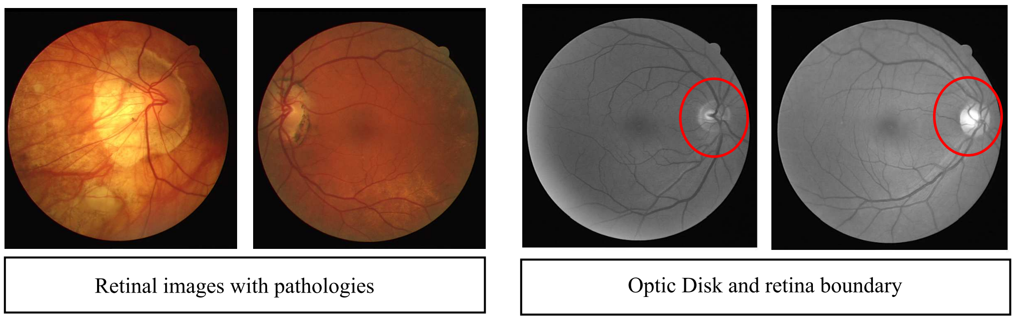
Retinal images with pathologies
Optic Disk and retina boundary

Fig. 1.6: *(Left)* Retinal images with pathologies; *(Right)* Retinal images showing the optic disk area and the retina boundary (marked with a red circle). Notice the contrast between that area and the background

These features have a high influence on the results obtained applying the previously commented techniques. In fact, among all the techniques presented in [Kirbas and Quek, 2004], active contours and neural networks have been shown as the most flexible tools for vessel extraction. On one hand, neural networks have been shown suitable to deal with medical images due to their ability to learn with a suitable training set, which should include

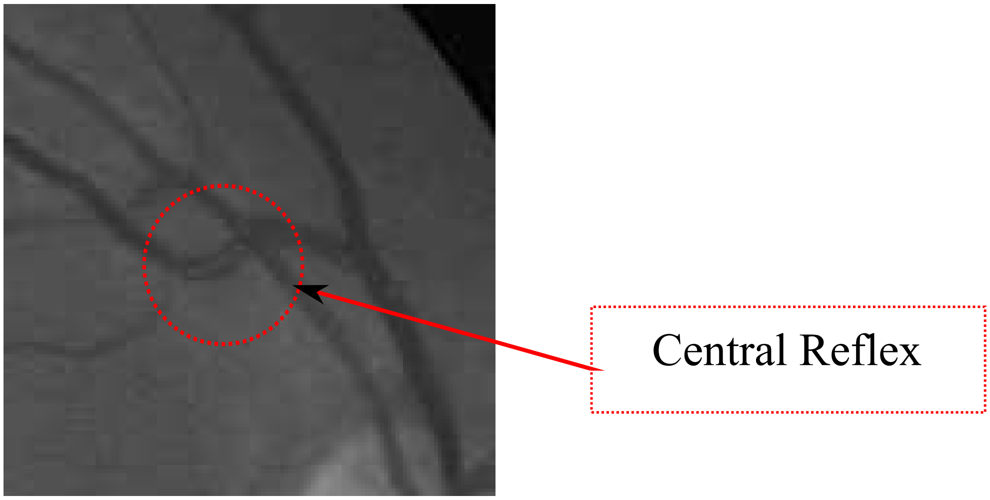

Fig. 1.7: Central reflex effect inside the vessel

all possible features or objects. However, every time a new feature is introduced to the network, a new training is needed. Another drawback is the fact that debugging the performance of the network is difficult. On the other hand, active contours have been also shown as a suitable and flexible tool for this task, 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 [McInerney and Terzopoulos, 1996b]. Active contours can manage reasonably noisy, indistinguishable or ambiguous object boundaries, quite common in medical images, like ultrasound images, angiographies, MRI or CT scans. Although this technique has been initially designed as an interactive tool [McInerney and Terzopoulos, 1995], some research has developed strategies trying to propose an automatic process for specific applications [Ourselin and Rongxin, 2005, McInerney and Terzopoulos, 1999, Valverde et al., 2001].

The future direction of segmentation research will be towards developing faster and more accurate and automated techniques, since the advances in radiological imaging systems produces a increasingly volume of patient images. A lot of research has been devoted to improve the computation speed by reducing the algorithm complexity [Toledo et al., 2000, Eviatar and Somorjai, 1996, Goldenberg et al., 2001, Hui et al., 2004]. Another way of achieving a faster computation is to use multi-scale processing technique which extracts the structures using low-resolution images, whereas narrow structures are extracted using high-resolution images [Li et al., 2006]. Alternatively, a pixel parallel approach is another way for a fast computation, assuming that a hardware device will exploit that massively parallelism.

1.3 Retinal vessel tree extraction proposal

A high variety of techniques have been developed to deal with the special features of retinal images. The current issue in the research is to improve the computation effort required maintaining a reasonable level of accuracy in the obtained results. As it has been previously commented, this improvement can be reached by means of reducing the complexity but at the cost of less accuracy.

Some retinal vessel processing techniques have been proposed in the literature intended to improve the execution time relying on a hardware implementation to exploit the massively parallelism. For example in [Perfetti et al., 2007], all the operations have been developed under the Cellular Neural Network (CNN) paradigm [Chua and Roska, 1993]. It assumes efficient processing in a pixel-parallel way. CNN constitute a class of recurrent locally coupled array of identical and simple processing elements. This fact has allowed the hardware implementation of a considerable number of processing elements into a chip. Further details of CNN paradigm can be seen in Appendix A. Although reliable results have been obtained, some of the proposed CNN-based operations utilise non linear templates which prevent their implementation in the current generation of cellular processor VLSI chips.

In this thesis, an algorithm for the automatically extraction of the retinal vessel tree is proposed (see Fig. 1.8). The retinal vessel tree extraction task is tackled with an active contour approach, based on the so-called topographic cellular active contours (TCAC) [Hillier et al., 2006], specially regarding on the computation effort. The TCAC originally appear intended to resolve the high computational cost of classic active contour techniques. They are based on a pixel-level discretisation of the contours and on a massively parallel computation on every contour cell which lead to a high-speed processing without penalising the contour location efficiency. In this case, the algorithm uses the Pixel Level Snakes (PLS) [Vilariño and Rekeczky, 2005], which represent an iterative TCAC technique where the contours evolve towards (local) minimal distance curves, based on a metric defined as a function of the features of interest. PLS are proposed in order to take advantage of the features of active contours and to alleviate the computation effort by means of exploiting the parallel nature of the hardware. PLS have been implemented on hardware architectures with capabilities of single instruction multiple data (SIMD) processing, like the cellular neural network-based chips ACE4K (under the ACE-BOX computing infrastructure) [Vilariño and Rekeczky, 2004b], as well as the focal plane processor array SCAMP-3 [Dudek and Vilariño, 2006a] and even on specific purpose integrated circuits [Brea et al., 2006]. PLS have demonstrated a good performance in some practical applications where active contours are frequently used, including segmentation and tracking of biological structures [Vilariño et al., 2004], moving object segmentation [Dudek and Vilariño, 2006a] and even for solving the shortest path problem in binary labyrinths [Vilariño and Rekeczky, 2004a].

Since the active contours have been initially designed as interactive models, a non-interactive application implies designing a suitable methodology in order to automatically initialise the contours close to the boundaries of interest. Note that fitting the interior of the vessels has been the most common technique used with active contour approaches for vessel extraction and segmentation [Caderno et al., 2004, McInerney and Terzopoulos, 1995]. However, that strategy has some disadvantages. In those cases where contours flow inside tubular structures, like the vessels, the internal potential stops the contour evolution due to its strength with respect to the external potential. So the evolution is controlled only by means of the external and balloon potentials. Moreover, complex rules should be defined to avoid the contour flowing outside the vessels [Caderno et al., 2004]. The initialisation process is also more complex, since fewer pixels belong to the vessels, around 12.7 % according to [Niemeijer et al., 2004].

In this thesis the strategy of fitting the exterior of the vessels is proposed in order to give more robustness to the PLS evolution and to simplify the definition of the initial conditions for PLS. Moreover, using the strategy of fitting the exterior of the vessels, the central reflex has no influence in the performance of the final solution. All the steps of the algorithm proposed in this thesis consist of simple local dynamic convolutions and morphological hit and miss operations together with simple arithmetic and logical operation, regarding its implementation in a processor array which will lead to a very fast computation speed.

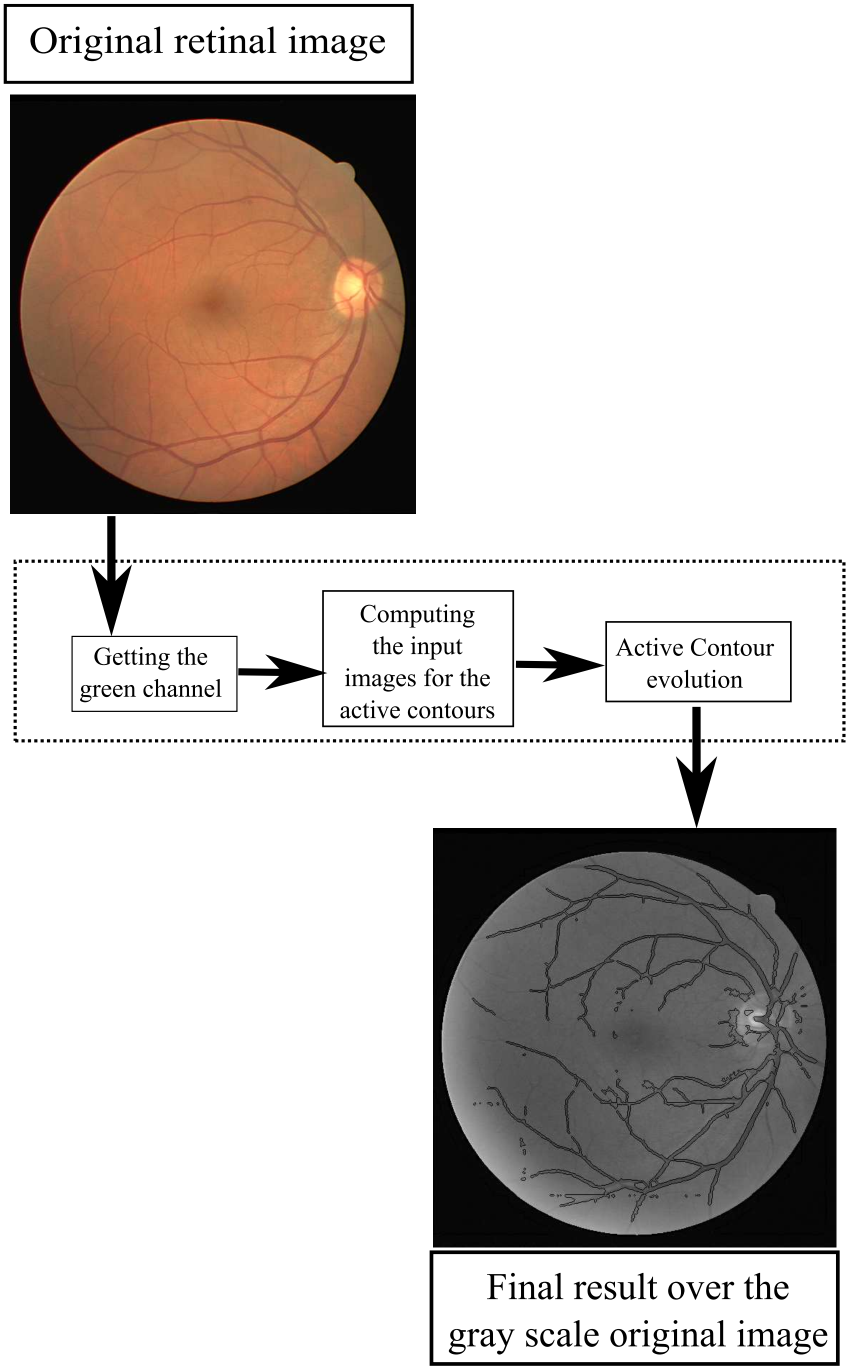
Original retinal image
Getting the green channel
Computing the input images for the active contours
Active Contour evolution
Final result over the gray scale original image

Fig. 1.8: Conceptual diagram of the proposed retinal vessel tree extraction algorithm

1.4 Outline

In *Chapter* 2 an introduction to the PLS performance is made. PLS features are detailed specially regarding on the retinal vessel tree extraction problem. In *Chapter* 3, the retinal vessel tree algorithm proposed in this thesis is described. In *Chapter* 4 is an analysis of the time performance and reliability of the obtained results of the algorithm. In *Chapter 5*, the results obtained with the algorithm are applied to several practical applications. Finally, *Chapter* 6 the main conclusions are stated.

Chapter 2

Pixel Level Snakes

As it has been commented in the introduction, active contours have been shown as effective tool for the medical images processing due to its robustness against noise. So, this technique has been considered to deal with the extraction of the retinal vessel tree in retinal images. One of the main disadvantages of active contour-based approaches is the high computation effort required. Pixel Level Snakes (PLS) [Vilariño and Rekeczky, 2005] have been proposed to tackle with the retinal vessel tree extraction task in order to take advantage of their massively parallelism feature improving the computation speed.

In this chapter, firstly, a brief introduction of active contours is made in order to conceptually define the basis of this technique. Then, PLS are explained in detail, regarding on the several implementations proposed in the literature as well as the specific performance related with the retinal vessel tree extraction task.

2.1 Active contours: an introduction

The classical active contour technique was firstly introduced by [Kass et al., 1988]. An active contour, also called snake, can be conceptually defined as an elastic curve that evolves from its initial shape and position as a result of the combined action of external and internal forces, see Fig. 2.1.

The parametric representation of the contour is described as a curve:

$$u(s) = (x(s), y(s)), \quad s \in [0, 1] \tag{2.1}$$

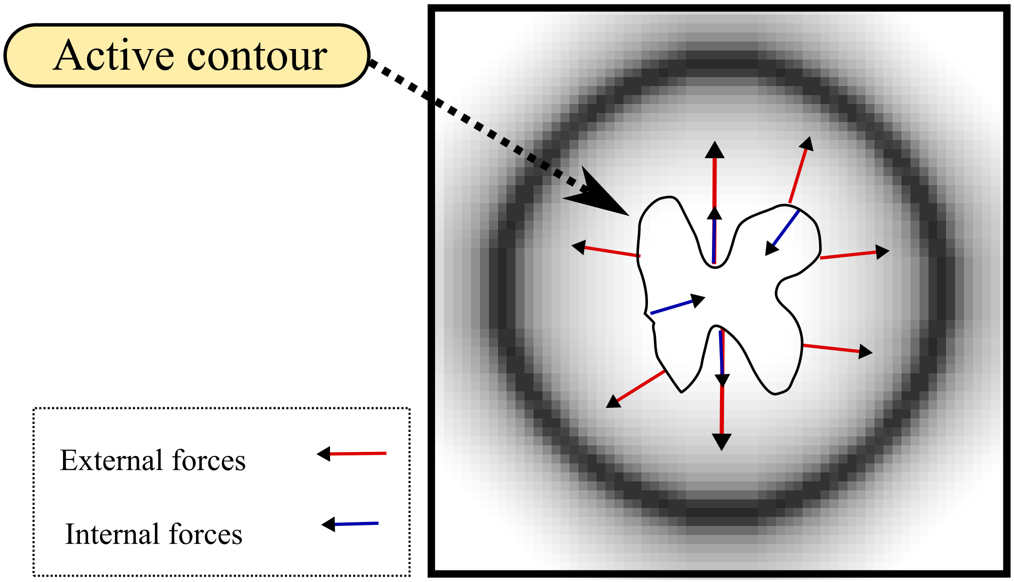

Fig. 2.1: Internal and external forces involved in the active contour evolution

where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain. The shape of the contour subject to an image $I(x, y)$ is associated with the function:

$$\varepsilon(u) = S(u) + P(u) \tag{2.2}$$

This function can be viewed as a representation of the energy of the contour in such a way that the final shape of the contour corresponds to the minimum of this energy.

The first term of Eq. 2.2 ($S(u)$),which corresponds to the internal deformation energy of the model, is defined as follows:

$$S(u) = \int_0^1 \left(\alpha \left| \frac{\partial u}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 u}{\partial s^2} \right| \right) ds \tag{2.3}$$

This term characterises the deformation of a stretchy, flexible contour. The parameters α and β control the tension and rigidity of the contour, respectively.

The second term of Eq. 2.2 ($P(x, y)$) corresponds to the external energy, which guides the contour evolution towards the boundaries of interest:

$$P(u) = \int_0^1 P_{ext}(u(s))ds \tag{2.4}$$

where $P(x, y)$ denotes a scalar potential function defined on the image plane. External potentials are designed in a way that a local minimum coincides with edges or other features of interest, depending on the problem under study.

The evolution of this model is basically a problem of minimising the energy function in Eq. 2.2. The contour $u(s)$ which minimises the energy function must satisfy the Euler-Lagrange equation:

$$-\frac{\partial}{\partial s}\left(\alpha\frac{\partial u}{\partial s}\right)+\frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 u}{\partial s^2}\right)+\nabla P_{ext}(u(s,t))=0 \qquad (2.5)$$

This vector-valued partial differential equation expresses the balance of internal and external forces when the contour rests at equilibrium. The first two terms represent the internal stretching and bending forces, respectively, whereas the third term represents the external forces. The evolution is controlled by all those forces in such a way that the internal forces are balanced with the external ones.

In order to numerically compute a minimum energy solution, a discretisation of the energy function $\varepsilon(u)$ is needed. The usual approach is to represent the continuous geometric model u in terms of linear combinations of local-support or global-support basis functions. Some of the proposals that can be found in the literature are finite differences [Kass et al., 1988], finite elements [Cohen and Cohen, 1993] and B-splines [Blake and Isard, 1999] methods.

Initially, this technique was designed as an interactive tool. In this sense, strategies have to be defined in order to properly initialise the contours. The management of topological changes, such as contour splitting or merging, cannot be handled. So, the number of the interesting regions and their approximate locations should be known *a priori*. Furthermore, highly irregular objects with deep and narrow cavities and/or protuberances cannot be approached. Moreover, the high computation effort required is the one of the drawbacks of this technique, depending on the application under study.

Some approaches have been proposed in the literature to overcome those limitations. Topology adaptive snakes [McInerney and Terzopoulos, 1995] (T-snakes) have been proposed in order to handle topological changes. This technique overcomes the topological inflexibility of classical active contours. This parametric snake uses a grid to re-parametrise the model during the deformation process. It is relatively independent of its initial position and is able to flow into complex shapes. However, the computation effort increases with the number of active contours. Moreover, topological changes are handled by means of an iterative process of evaluation and re-parametrisation.

Since the energy estimation is strongly dependent on the parametrisation, it changes in each of the intermediate re-parametrisation steps, allowing to reach tubular and/or irregular shapes but at the cost of a penalisation in the internal energy performance.

Topological changes are managed in a simpler and more natural way by the propagating technique proposed by Osher and Sethian [Osher and Sethian, 1988], the Level Set Method (LSM). The main idea behind the LSM is to represent propagating curves as the zero level set of a higher dimensional function which is given in the Eulerian coordinate system. Hence, a moving wave front is captured implicitly by the level set function (LSF). The main advantages of this approach are:

- Complex shapes can be properly handled, since it develops sharper corners and change their topology during the evolution
- Intrinsic properties of the propagating wave front such as the curvature and the normal to the curve can be easily extracted from the LSF
- It is easily extendable to higher dimensions
- Since the LSF is given in the Eulerian coordinate system, discrete grids can be used to obtain a numerical approximation to the solution

Similar approaches based on propagation waves under a curvature dependent speed function are the geodesic and wave front propagation proposed in [Caselles et al., 1997, Malladi et al., 1995], respectively. In all those strategies, the main difficulties are the constrains imposed due to the higher dimensionality of the embedding hipersurface and the implicit definition of the contour model. Moreover, a remarkable number of specific terms should be defined to control the evolution, such as propagation speed and suitable stop criteria.

The main drawbacks of all these proposals is still the required computation effort, since a compromise between the processing speed and flexibility in the contour evolution is needed. As a difference with the commented strategies the topographic cellular active contours (TCAC) are intended to solve the high computational cost inherent to the classical active contours techniques taking advantage of the massively parallelism. Up to the present, two different cellular active contour approaches have been proposed. In [Rekeczky and Chua, 1999] an active wave computing approach is introduced. This consists of a topographic non-iterative region propagation technique where the contours are defined by the boundaries of trigger waves. Therefore, as in the level-set approaches, the contour evolution is implicitly represented as a wavefront propagation. This approach has demonstrated a high flexibility in the contour evolution, and a simple solution is given to the changes of topology, required when two different wavefronts collide. However, sophisticated stop criteria are usually required to control the wavefront propagation, increasing significantly the computation complexity in real applications.

Pixel Level Snakes (PLS) represent a topographic iterative active contour where the contours are explicitly represented and evolve towards local minimal distance curves based on a metric defined as a function of the boundaries of interest. PLS are situated in the middle way between parametric and implicit models. Due to their high level of discretisation and the characteristics of evolution (pixel to pixel) they can handle topologic transformations by operations perfectly integrated and accessible in the evolution process. This also allows to control the topologic transformations by preventing the collision between contours in a simple way. They can also delimit objects with cavities and protuberances keeping a high control of the contour shape due to the definition of deformation potential terms based on the curvature which is locally estimated.

2.2 Pixel Level Snakes

Pixel Level Snakes (PLS) are based on a pixel-level discretisation of the contours and on a massively parallel computation on every contour cell. PLS contours are guided by local information and regularising terms dependent on the contour curvature, similar to the geodesic active contours. An explicit evolution is made, like in the parametric technique, allowing more control over the topologic features of the contours.

PLS can be conceptually decomposed by three different modules which interact dynamically: *guiding information extraction, contour evolution and topological transformations* modules (see scheme in Fig. 2.2).

The PLS inputs consist of a binary image containing a suitable initialisation, a multi bit image (external potential image) containing guiding information from the image under processing, and finally, some parameters to control the contour evolution. The *guiding information extraction module* extracts the information to guide the contour evolution towards the boundaries

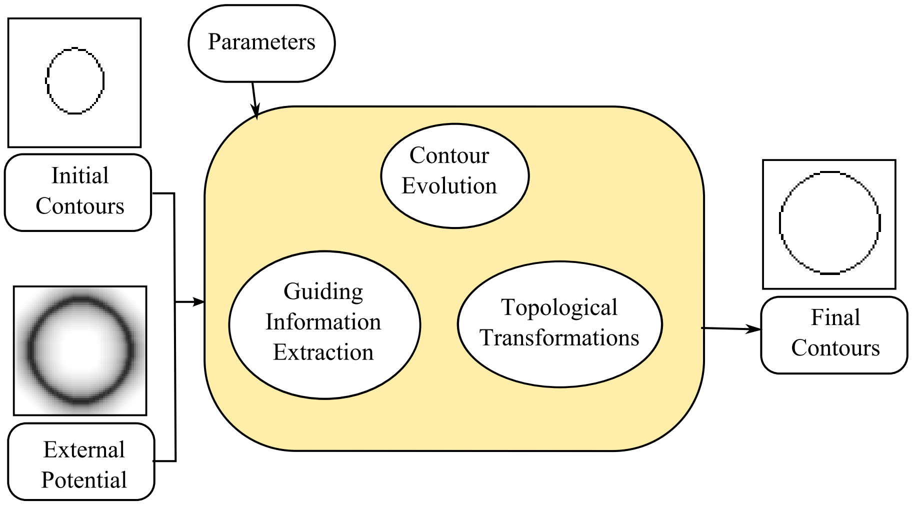

Fig. 2.2: PLS schema. These three modules interacts dynamically along the four cardinal directions (North, South, East, West)

of interest. The *contour evolution module* consists of an iterative operation of pixel-to-pixel shift of the contours, controlled by the guiding information image. The *topological transformation module* handles the changes in the topology, such as contour splitting and merging, as well as the preservation of the topology, when it is required.

In the PLS formulation, the active contours are represented as sets of 8-connected activated pixels in a binary image, called contour image, with the same dimensions as the original image. The contour evolution consists on an iterative process based on binary and local morphological operations along the four cardinal directions. The goal after each cycle is to obtain a new well-defined contour slightly shifted and/or deformed in order to be closer to the boundaries of interest. These operations are guided by external forces, extracted from the image under processing, and internal forces, derived from the contours themselves. The guiding force, which control the contour evolution, is defined based on a potential field ($P(x, y)$):

$$\vec{F} = -\nabla P(x, y) = -\frac{\partial P}{\partial x}\vec{i} - \frac{\partial P}{\partial y}\vec{j} = F_x\vec{i} + F_y\vec{j} \qquad (2.6)$$

So, the guiding forces for the contour evolution are extracted from a global potential, which will be a combination of the internal and external potentials:

$$P(x, y) = k_{int}P_{int}(x, y) + k_{ext}P_{ext}(x, y) \qquad (2.7)$$

where k_{int} and k_{ext} are real and positive coefficients which weight the influence of each kind of information. The combination of both external and internal potentials would provide more robustness to the contour evolution against noise. These values must be determined heuristically, like in classical active contour techniques. According to this formulation, only the sign of the component of the guiding forces along the direction under exploration is actually needed. The external potential is defined in such a way that the boundaries of interest coincide with the valleys of the potential field. In Fig. 2.3, the evolution using only external potential is shown. Notice that the active contour fits the circular shape of the external potential image flowing towards the black pixel intensities.

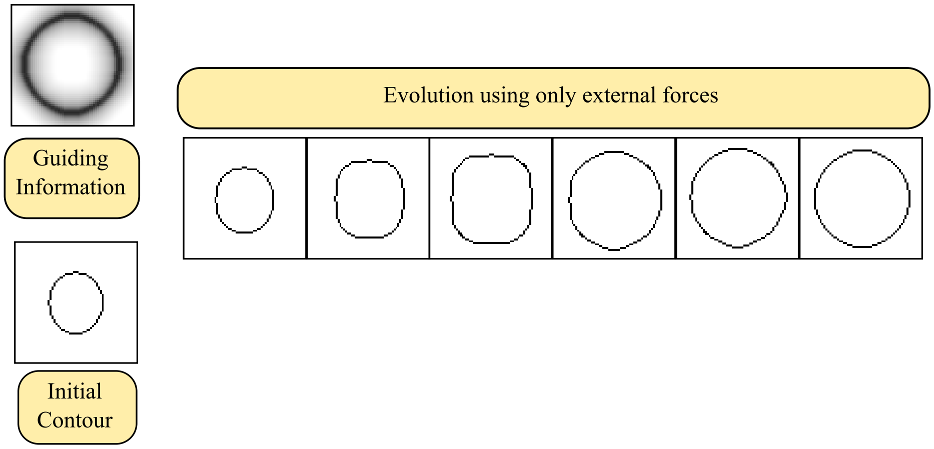

Fig. 2.3: Contour evolution using only external forces. Notice that the potential value coincides with the pixel intensity

The internal potential is derived directly from the active contours. In classical approaches, this term depends on the tension and flexion of the contour and it may be measured as a function of distances among adjacent points according to the considered discretisation. This approach cannot be directly included in the PLS formulation, because the contour is not defined as a predetermined number of discretisation points but as a variable set of black pixels of a binary image. However, the desired smoothing effect can be obtained by assigning higher potential values for those pixels in the contour image situated outside of the contour cavities, with respect to those situated inside. One way to perform this is by means of a recursive low-pass filtering or diffusion operation acting on the contour image. A directional gradient operation acting on this array will originate positive internal forces which push to reduce the local curvature, smoothing the contour shape. This idea is shown in Fig. 2.4 where the contour is guided only by this kind of internal forces. A closed contour whose evolution relies only on the local curvature will adopt a circular shape and finally collapse.

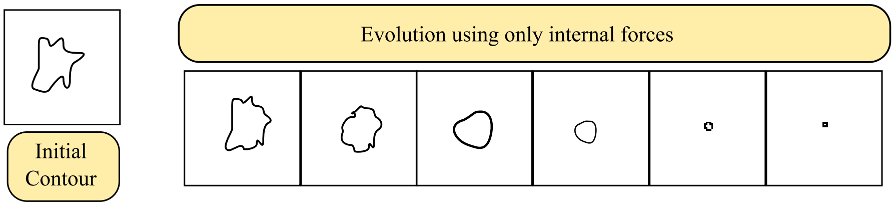

Fig. 2.4: Contour evolution using only internal forces

PLS also includes balloon forces, a kind of internal potential, which are intended to improve the robustness and stability of the PLS evolution. The balloon forces are originally defined as follows [Cohen and Cohen, 1993]:

$$\vec{F} = k_1 \vec{n}(s) \tag{2.8}$$

where $\vec{n}(s)$ is the unit vector normal to the curve $u(s)$ and k_1 is a constraint which controls the inflation or deflation tendency (depending on the sign of k_1) respect to the external forces. PLS can effectively inflate or deflate the contours by the definition of the new potential field as follows:

$$P_{inf} = s_{inf} * \xi(x, y) \tag{2.9}$$

where $\xi(x, y) > 0$ if (x, y) belongs to the set of locations enclosed by an active contour, otherwise, $\xi(x, y) < 0$, and finally, $s_{inf} = +1$ defines an inflating potential and $s_{inf} = -1$ a deflating potential. The global potential field is redefined as follows (compare with Eq. 2.7):

$$P(x, y) = k_{int} P_{int}(x, y) + k_{ext} P_{ext}(x, y) + k_{inf} P_{inf}(x, y) \tag{2.10}$$

The balloon potential is weighted by the real and positive coefficient k_{inf}. This potential will move the contours in those cases where the external potential is too weak. The balloon forces can help the contour to remove spurious noise, and counteract their tendency to shrink (due to the internal forces). In Fig. 2.5 is shown the evolution of the contour using only an expansion potential.

From the point of view of either the image processing or the hardware implementation, the PLS technique has undergone several modifications in order to improve its performance [Vilariño and Dudek, 2007, Vilariño and Rekeczky, 2005]. Conceptually, PLS can be classified into contour and region based approaches. In the following sections, the performance of those approaches will be discussed.

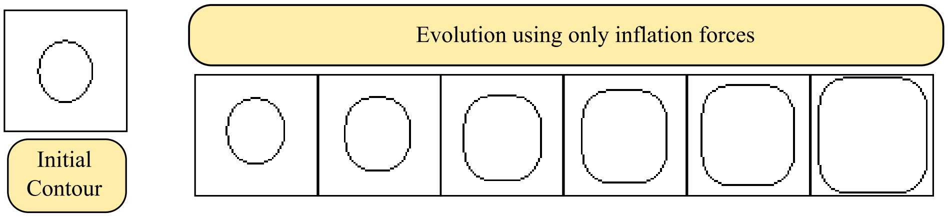

Fig. 2.5: Contour evolution using only balloon forces, in this case an inflation force is shown

2.2.1 Contour based PLS

The first fully operative PLS implementation was proposed in [Vilariño et al., 2000]. A new proposal was published in [Vilariño and Rekeczky, 2005] in order to improve the computation effort and the active contour performance of the first version. The flow diagram for the contour based PLS version can be seen in Fig. 2.6.

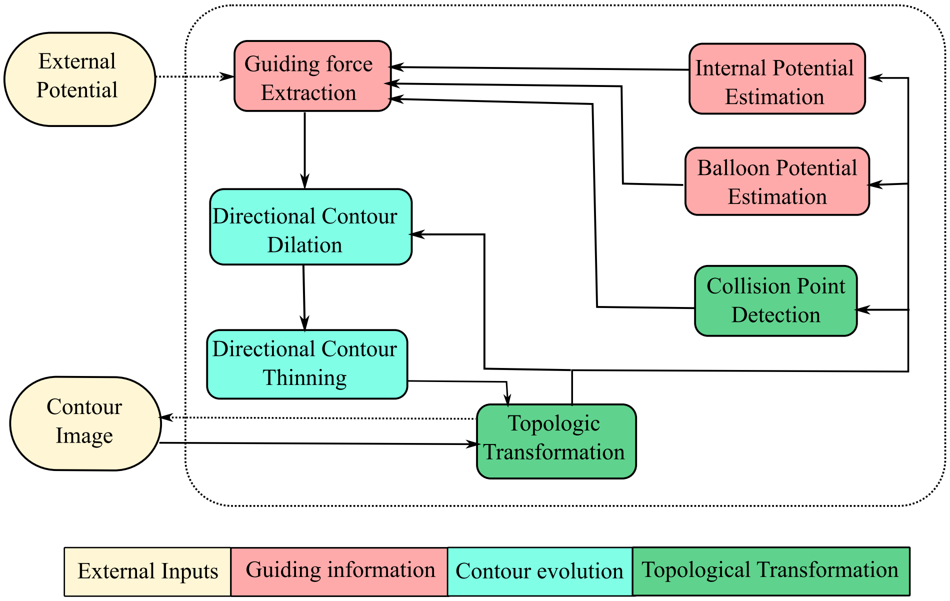
External Potential
Guiding force Extraction
Internal Potential Estimation
Balloon Potential Estimation
Directional Contour Dilation
Collision Point Detection
Directional Contour Thinning
Contour Image
Topologic Transformation
External Inputs
Guiding information
Contour evolution
Topological Transformation

Fig. 2.6: Flow diagram of the modules defined for the contour based PLS

The contour evolution is controlled by means of both the *directional contour dilation* (DCD) and the *directional contour thinning* (DCT) modules, which dilate and thin the contours, respectively, along the direction of processing. The evolution is based on the activation and deactivation of pixels of the binary contour image. The sign of the component of the guiding forces, which controls the evolution, is extracted in the so-called *guiding force extraction module* (GFE). The output of this GFE module will represent a binary map with activated pixels in those locations where the potential is decreasing along the direction under study. The internal potential is derived directly from the active contours, in the *internal potential estimation* (IPE) module. This operation is performed by means of a recursive low-pass filtering or diffusion operation acting on the contour image.

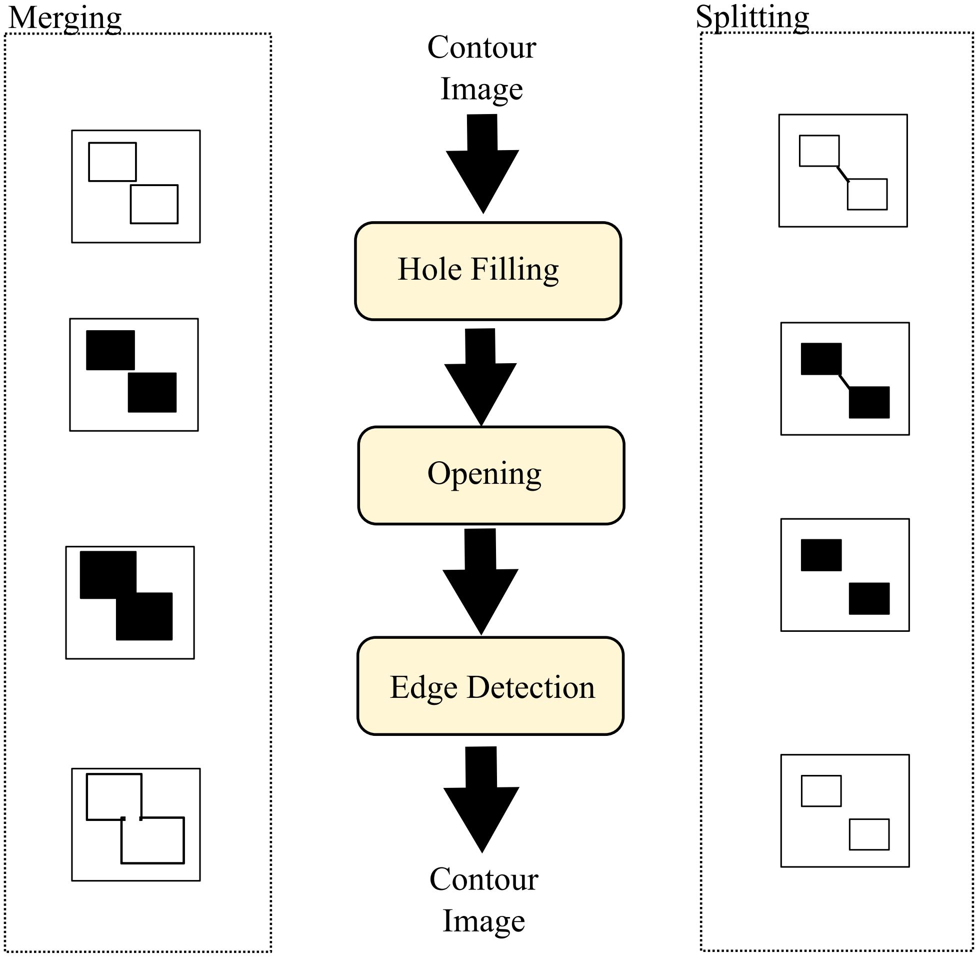

Fig. 2.7: Flow diagram of the operations in the topologic transformations module, for the contour-based PLS

Changes of topology can be classified as follows:

- avoiding uncontrolled collisions between contours to preserve the topology

- splitting and merging contours

These operations are managed in the *collision point detection* (CPD) module. Avoiding the collision between contours relies on a pre-estimation of the locations where a collision could occur. The CPD is carried out by a simple pattern recognition which takes as input the binary contour image and returns a binary image with deactivated pixels in those locations where a collision between contours can appear in the next iteration. Therefore, by the projection of this binary map onto the output of the GFE module, the pixel activation can be avoided on those conflictive locations and consequently the contour collision will be prevented. Splitting and merging of contours are based on real collision points. In this case, the contours are defined as frontier pixels of regions into the image space. Initially, the associated regions were obtained from the contours by means of a hole filling operation followed by a one-step morphological opening (erosion followed by dilation), see Fig. 2.7. In the last step the region contours were obtained by a binary edge detection which extracts the set of frontier pixels of the region. This strategy handles properly the changes of topology whenever they are required. However, the inclusion of the hole filling operation to get the area surrounded by contours penalises the computational effort in architectures based on iterative processing, since it is performed in each cycle of the evolution process. Moreover, this operation introduces an important constrain in the image processing, since it is not possible to manage active contours evolving inside of other contours.

Some modifications were proposed to avoid the intensive use of the hole filling operation [Vilariño and Rekeczky, 2005]. The hole filling operation was replaced by local non-propagative operations to update the regions associated with the contours after each iteration, instead of computing them only from the contour images. In each iteration, the image contains the regions surrounded by the active contours of the previous iteration, which simplifies the execution of the topologic transformations as well as the estimation of the balloon potential.

2.2.2 Region based PLS

Another approach is to define PLS implicitly as the boundaries of active regions, see flow diagram in Fig. 2.8 [Dudek and Vilariño, 2006b].

This approach follows the general implementation of the PLS algorithm, but is based on the evolution of an active region. The contour is defined

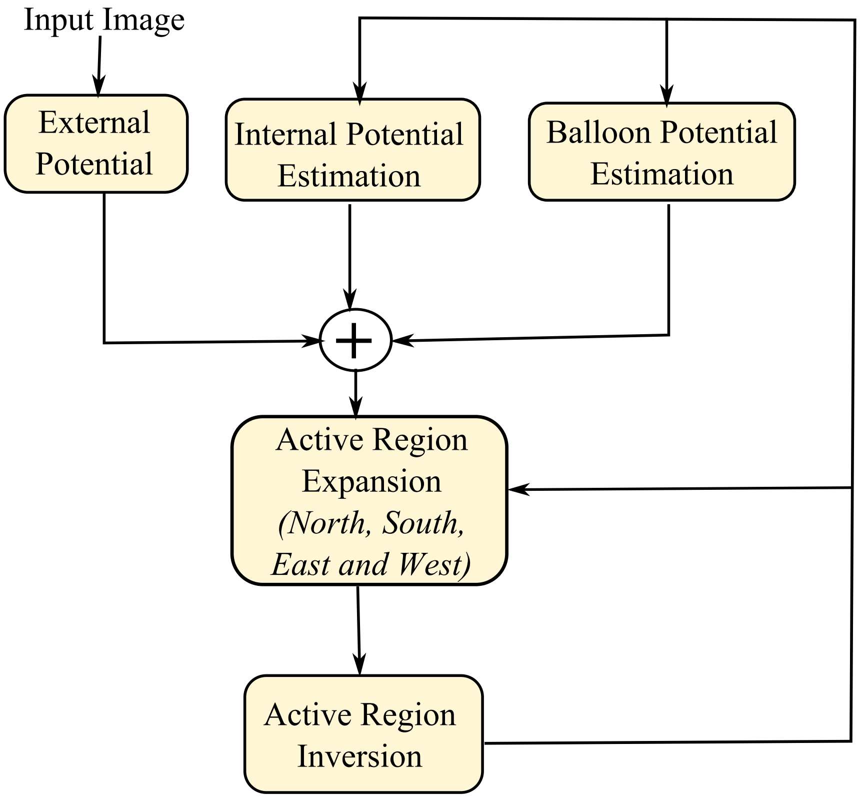
Input Image
External Potential
Internal Potential Estimation
Balloon Potential Estimation
+
Active Region Expansion *(North, South, East and West)*
Active Region Inversion

Fig. 2.8: Flow diagram of the region-based PLS version

implicitly as the boundary of the region. This fact simplifies not only topographic transformations (merging and splitting of contours); but the implementation of inflating/deflating forces is also easier. The main advantage of this approach if the elimination of the hole-filling step, which is time-consuming when it is performed iteratively on a processor array.

The contour evolution is implemented using very simple local rules, which results in a fast implementation. This contour evolution is implemented through conditional expansion of active regions in four cardinal directions (North, South, East, West). This is followed by an inversion of the active regions and another conditional expansion of the new active regions. In this way, the contours can move in every direction, and follow the guiding forces to settle into the minimum potential location. A global cycle of eight movements is actually needed to perform an iteration in the PLS evolution. The internal potential is estimated by gray-level diffusion of the contour image which is implicitly associated with the active regions. The balloon potential is simply obtained by multiplying the binary active region images with the balloon potential weight. This will define an inside/outside area of the balloon. The global potential field is obtained by adding of all the potentials, the external, internal and balloon potentials. This sum provides a map with activated and deactivated pixels controlling the contour evolution towards the selected direction.

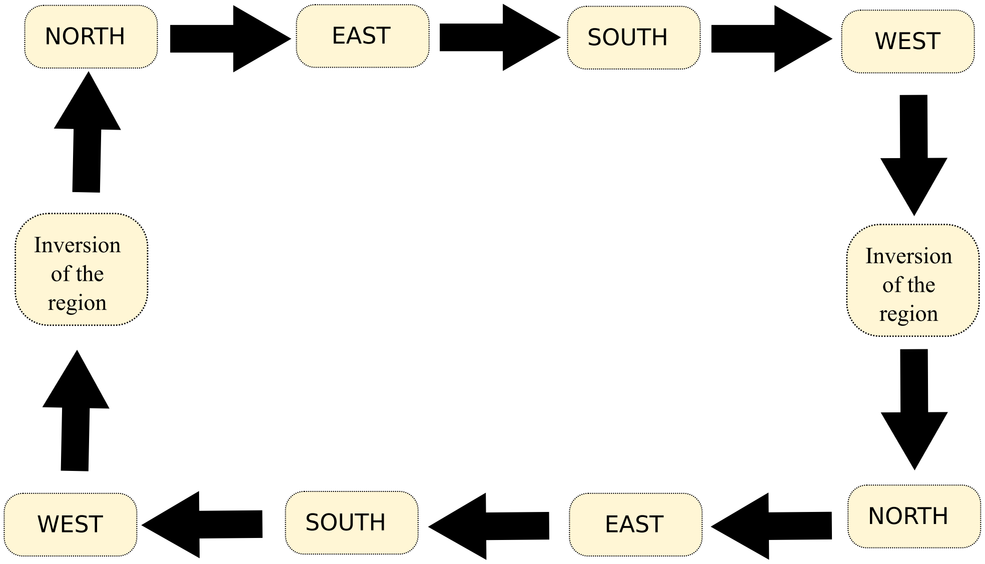

Fig. 2.9: Number of steps required for an iteration in a cycle of the PLS evolution

The active region inversion allows the implementation of inflation and deflation forces, allowing the movement of PLS in every direction. So, the global number of steps required for an iteration of the PLS cycle is eight, see Fig. 2.9. As the active region expands, it is possible that it merges with another region. The topologic transformations (such as merging or splitting) are achieved through implicit definition of the contour, and do not require any additional computations (see Fig. 2.10). In some applications, it is important to prevent the contours from colliding with each other. This is implemented by means of detecting possibility of collision of each pixel from an active region in every direction. So, if there is a collision the directional expansion is modified to avoid it and to preserve the topology.

Since it is needed that the contours can move in every direction, the active region is inverted before the next iteration of the algorithm loop. The inversion of the active region results in a new contour, defined implicitly as lying inside the active region. Then, the contour evolution is performed again. So, the PLS cycles is defined through expansions and contractions of the active regions. In each inversion of the active region, the balloon potential is also inverted to maintain the consistent direction of the inflating/deflating forces.

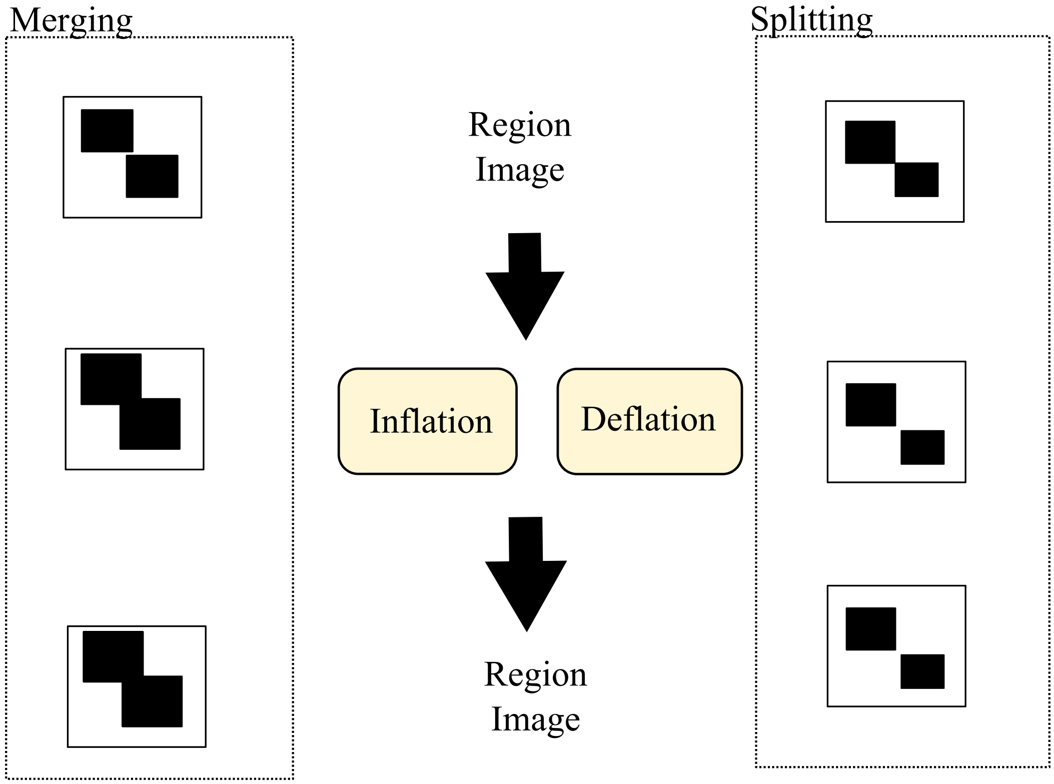

Fig. 2.10: Merging and splitting for the region-based PLS

Initially, region based PLS seem to be simpler than previous versions. However, the number of steps is not clearly lower because a global cycle, along the four cardinal directions requires eight steps with the algorithm against only four of the previous PLS algorithms [Dudek and Vilariño, 2006b, Vilariño and Dudek, 2007]. Another possible disadvantage depending on the task is the fact that nested contours (closed contours within other contours) as well as propagation inside 1-pixel wide cavities are enabled. This feature implies the necessity of a more accurate guiding information image in order to properly stop the evolution.

2.2.3 Discussion

Different PLS approaches have been discussed in this chapter. Although the general implementation is maintained in all the versions, their performance is slightly different providing advantages and disadvantages depending on the task under study.

In this thesis, the aim is to develop an algorithm to fit the exterior of the vessels using PLS, and calibrating them for this specific task.

Initially, the contour based PLS version has been proposed to tackle with the retinal vessel tree extraction task. This allowed to robustly control the PLS evolution, specially in those cases where small discontinuities appears in the vessel topology quite common after the processing of the retinal images due to vessel ambiguous boundaries. On the contrary, the region-based PLS version can flow through 1-pixel-width cavities, which implies the necessity of a more accurate external potential image to control the evolution. Initially, contour based PLS could seem to be faster since in each evolution cycle only four movements are needed, instead of the eight needed in the region based PLS version. Nevertheless, unlike the contour based version, the region version can be easily adapted for a custom application optimising the execution time required. In this sense, since the algorithm strategy consists on fitting the exterior of the vessels, PLS will experiment only expanding evolution. So, the number of PLS iterations can be reduced to four, which implies a faster computation.

Chapter 3

Pixel parallel retinal vessel tree extraction algorithm

In this thesis, a retinal vessel tree extraction algorithm has been proposed in order to automatically fit the exterior of the vessels in retinal images, by means of the pixel parallel active contour technique, the so-called PLS described in the previous chapter. The main goal is to automatically compute the input images needed by PLS, relying only on the local statistics of the original retinal image. Moreover, the calibration and tuning of the PLS parameters is needed for a robust and optimised performance of the retinal vessel tree extraction. Notice that the strategy presented in this thesis consists on fitting the exterior of the vessels instead of the interior. As it has been previously commented, fitting the interior of the vessels has some disadvantages. Contour evolution is only controlled by external and balloon forces, since the internal potential should be disabled. The management of the contour evolution inside the vessel is more complex. In this sense, complex strategies have to be defined to avoid flowing outside the vessel locations. The central reflex can also affect the evolution. Moreover, the initialisation when contours are fitting the interior of the vessels is quite more complex due to the less proportion of foreground pixels than background ones.

Conceptually, several stages have been defined for the retinal vessel tree extraction, see Fig. 3.1. Initially, a *vessel pre-estimation* stage has been proposed to approximately determine the vessel edge locations with some pre-filtering steps to improve the signal-to-noise ratio. Then, the initial conditions and the external potential images are computed in the *second and third Stages*, using the output images from the previous stage. Finally, PLS will evolve to fit the exterior of the vessel edges using both of the previously computed images.

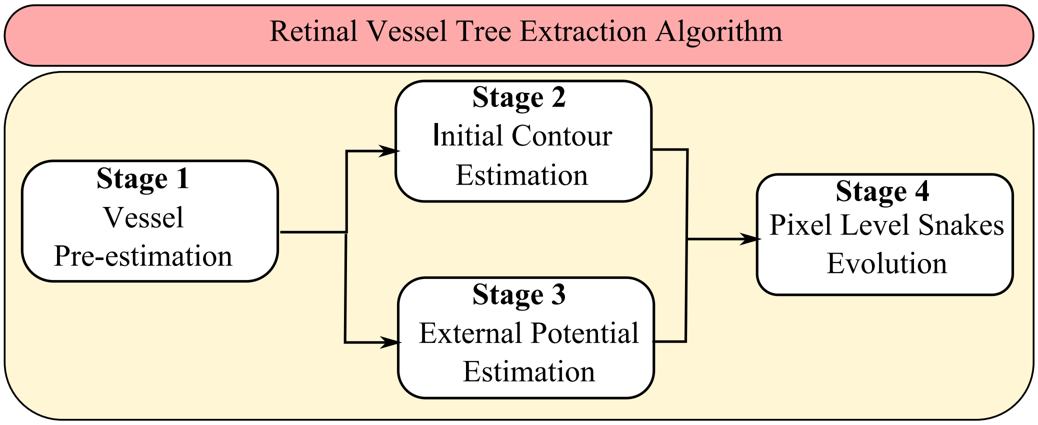

Fig. 3.1: Flow diagram showing the main stages defined for the retinal vessel tree extraction algorithm proposed in this thesis

One of the main motivations of this proposal is the improvement of the computation time required taking advantage of the massively parallel processing of the pixel-parallel hardware devices. The first attempt to solve the task consisted on implementing the conceptual stages using steps defined in terms of local operations and convolutions together with arithmetic and logical operations in order to be implemented in a cellular processor array. Initially in the first version of the algorithm, all the operations and steps of the methodology were defined under the CNNUM paradigm [Chua and Roska, 1993]. The contour based PLS addressed in [Vilariño and Rekeczky, 2005] has been used in the original version. Nevertheless, the complexity of some steps prevented their implementation in the current generation of cellular chips. In order to solve this problem, some modifications of the algorithm were proposed to make easy the hardware implementation and they were presented in the final approach of the algorithm. Moreover, this final version was optimised and custom implemented for the retinal vessel tree extraction task to make a faster computation.

Due to the high resolution of the retinal images, original retinal image has been split into NxN sub windows in order to fit the size of processor arrays without losing the image information.

In the following sections, the original and final approaches of the retinal vessel tree extraction algorithm proposed in this thesis are presented. All the steps of the stages are detailed and defined. Finally, the main conclusions are discussed.

3.1 Original approach: CNN-based retinal vessel tree extraction

Obviously, the first approach was to solve the retinal vessel tree extraction task from an image processing point of view; but regarding the use of local convolutions, arithmetic and logical operations for a future projection onto a pixel parallel processor array. For this first attempt, all the steps used to define the stages were defined using CNN-based operations, particularly those defined in the CNN Software Library (CLS) [CLS, 2007], see Fig. 3.2. This proposal was published in [Alonso-Montes et al., 2005a,b]

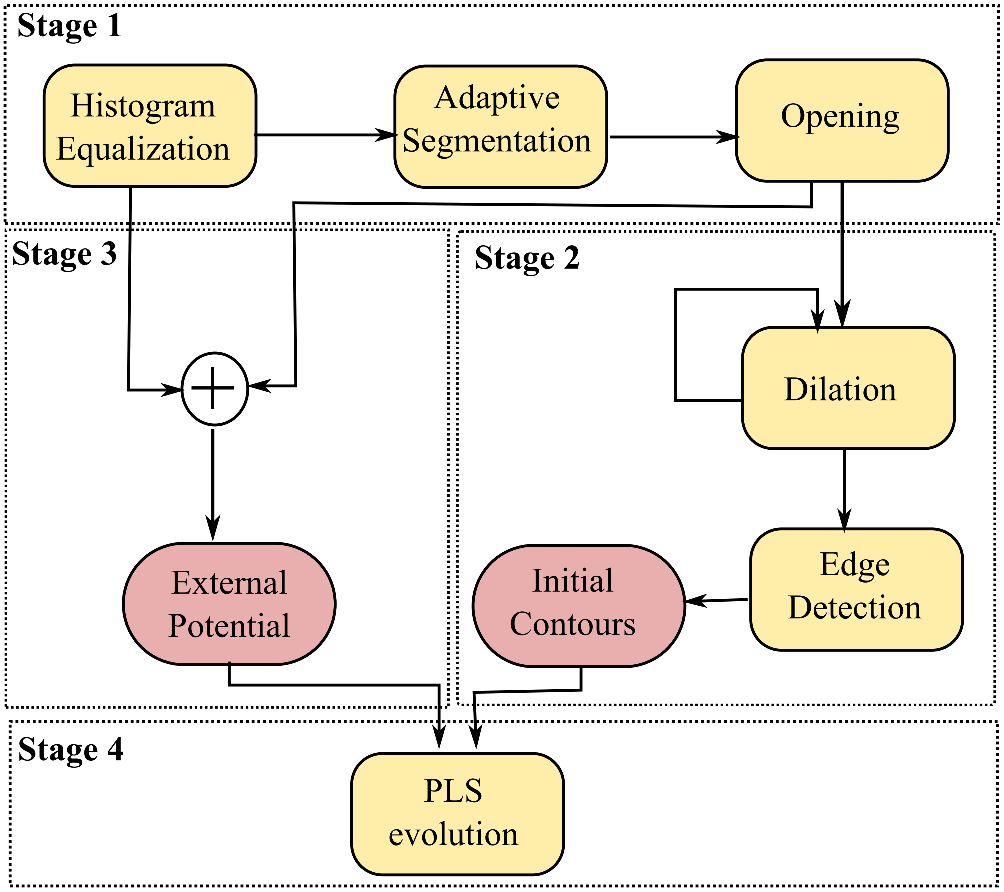

Fig. 3.2: Flow diagram of the original algorithm using CNN-based operations

Although all the steps can be implemented using conventional image processing techniques, this CNN-based implementation was proposed to take advantage of the computation time reduction provided by the massive parallel processing.

3.1.1 Stage 1. Vessel Pre-estimation

The main goal of this stage is to pre-estimate the regions between the vessels in order to be used in the following stages for the estimation of the initial condition and the external potential images. Several steps were proposed to get the pre-estimation of the vessel regions, see Fig. 3.3.

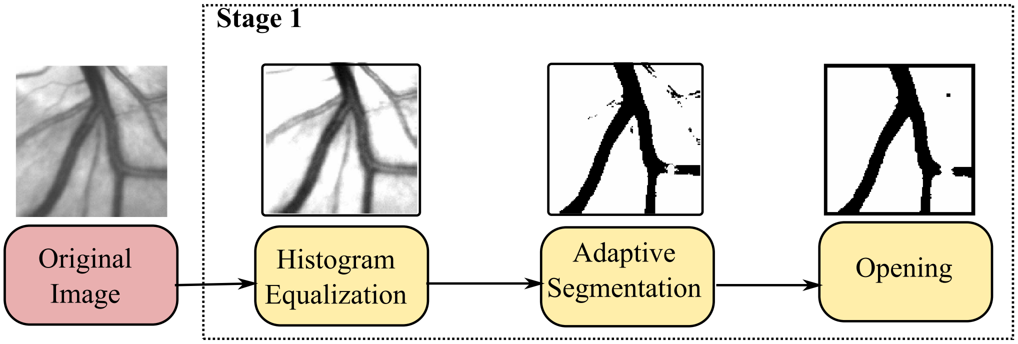

Fig. 3.3: Stage 1: The original image is equalised, and then an adaptive segmentation followed by an opening step are applied to get the segmented image

First, an histogram equalisation was proposed to improve low contrast vessels. Due to the high gray level variability along the vessels, the CNN-based adaptive segmentation addressed in [Rekeczky et al., 1997] was proposed for this step. It consists on a local threshold estimation followed by a locally adaptive segmentation. The local threshold estimation determines the space-variant threshold level (T_{est}) computed from local statistics:

$$T_{est} = \alpha E_m + \beta E_v + thres, \quad \alpha \in [0,1], \beta \in [-1,0] \tag{3.1}$$

where E_m and E_v are the mean and the variance estimations of the considered image, respectively; $thres$ is a constant threshold value which depends on the gray-level of the considered image and α and β are scale factors, whose values are heuristically estimated. The motivation behind the formulation proposed in Eq. 3.1 is to establish an optimal threshold which better describe the object boundaries. An estimation based only on the average in a local neighbourhood is not enough for the optimal separation of the objects from the background. In this sense, the variance is significantly higher at the boundaries than in homogeneous regions. The combination of both, the mean and the variance values, makes more efficient the estimation of T_{est} and improves the segmentation result. The mean in the boundary should be decreased $(\alpha > 0, \beta < 0)$ to improve the threshold estimation. Taking into account the type of retinal images used in our algorithm, the *thres* value is proposed to be computed as follows:

$$thres = \max\left[\frac{\sum_{i=1}^{N} I_{i1}}{N}, \frac{\sum_{i=1}^{N} I_{i2}}{N}, \quad \ldots \quad, \frac{\sum_{i=1}^{N} I_{iM}}{N}\right] \tag{3.2}$$

where I_{ij} is the gray-level value in the i-row and j-column in the NxM input image and $(\sum_{i=1}^{N} I_{ij})/N$ is the mean value of the j-column. The idea behind this formulation is to establish a value according to the mean values of the image. Since the proportion of pixels from the background is bigger, and the vessels always have a high gray level variability, the mean value of each column gives a mean value, which is closer to the vessel. The maximum selection allows us to get a suitable compromise between the noise and the segmented vessel edges, emphasising the difference between foreground (vessel points) and background.

Finally, a morphological opening is applied to eliminate noisy points, maintaining at the same time, the vessel topology. As result, an estimation of the vessel locations is obtained.

3.1.2 Stage 2. Initial Contour Estimation

The aim of this stage is to get a suitable initialisation for the PLS in order to fit the exterior of the vessels. Several steps were proposed to deal with this task (see Fig. 3.4).

Since our strategy consists on fitting the exterior of the vessels, the goal was to assure that the initial conditions computed for the PLS are completely outside from the vessel locations. The segmented image obtained in the first stage is used as the starting point to compute the initialisation needed by PLS since it provides a good pre-estimation of the actual vessel locations. Due to the central reflex, the interior of the vessels is not properly segmented. So, several dilations are made to avoid those problems. Thus, the obtained image contains only regions situated completely outside of the the vessel locations. Finally, a binary edge detection is made to get the contours used as input for PLS.

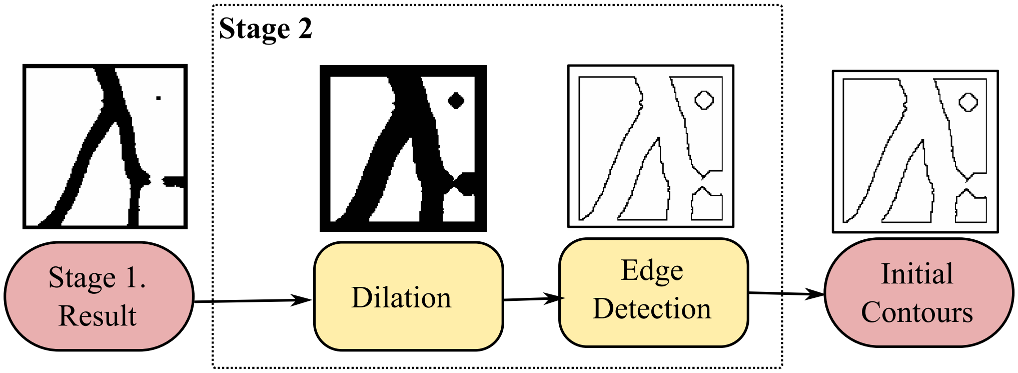

Fig. 3.4: Stage 2: Several dilations and a binary edge detection allows to get the initial locations needed by PLS

3.1.3 Stage 3. External Potential Estimation

The goal in this stage is to compute the external potential image needed by PLS to guide the evolution towards the vessel edges. The images obtained during the 1st stage were combined and weighted to compute this external potential image (see Fig. 3.5).

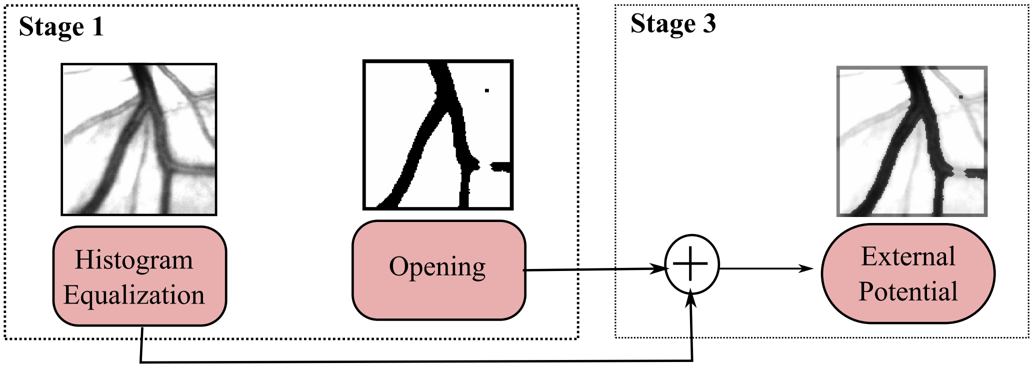

Fig. 3.5: Stage 3: The images from 1st stage are used to compute the external potential image

The equalised and segmented images from the *Stage 1* are combined as follows:

$$I_{ext} = \rho I_{eq} + \delta I_{op} \tag{3.3}$$

where I_{eq} and I_{op} corresponds to the equalised and the segmented images, respectively, ρ and δ are scale factors. I_{eq} contains the segmented vessels, whereas I_{eq} gives additional information regarding the vessel topology, emphasising the edges. Noise is at the same time smoothed.

3.1.4 Stage 4. PLS evolution

In this first approach, the contour-based PLS [Dudek and Vilariño, 2006b] was used as a general approximation to the problem. The main issue of this stage consisted on tuning and calibrating the main parameters which control the PLS evolution. The internal potential is used to prevent the evolution through the vessel discontinuities maintaining the vessel topology. An inflation potential should be established to move the contours in those cases where the external potential is too weak (see Fig. 3.6). Notice that in the contour-based approach the hole filling operation removes noisy points associated with residual contours, but at the cost of increasing the computation effort.

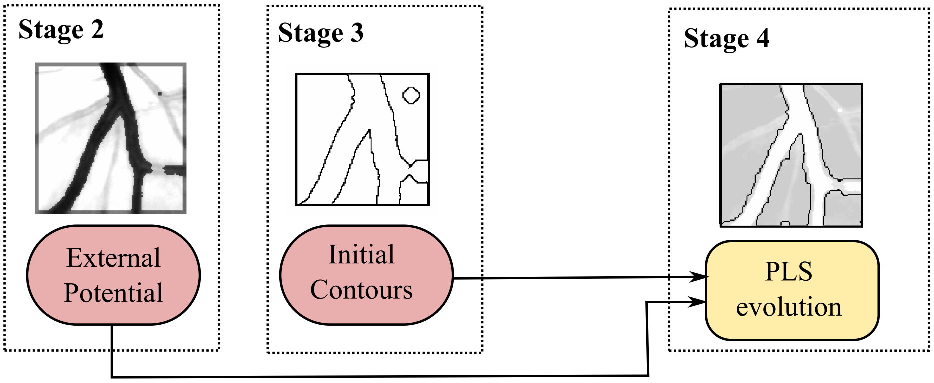

Fig. 3.6: Stage 4: Both of the previously computed input images are used by PLS to evolve and fit the exterior of the vessels

Appendix B shows the experimental results obtained using this first version of the algorithm. Although good results were obtained, the main drawback of this proposal is the fact that the adaptive segmentation step cannot be implemented in the current generation of processor arrays due its complexity.

3.2 Final approach using region-based PLS

A final approach was proposed in [Alonso-Montes et al., 2007, 2008c] in order to overcome the limitations of the previous version. The main issue was to define all the steps in such a way that they can be implemented in a current SIMD processor array. The stages were modified (see Fig. 3.7) to fit the specific requirements of PLS and to propose a full hardware implementable algorithm.

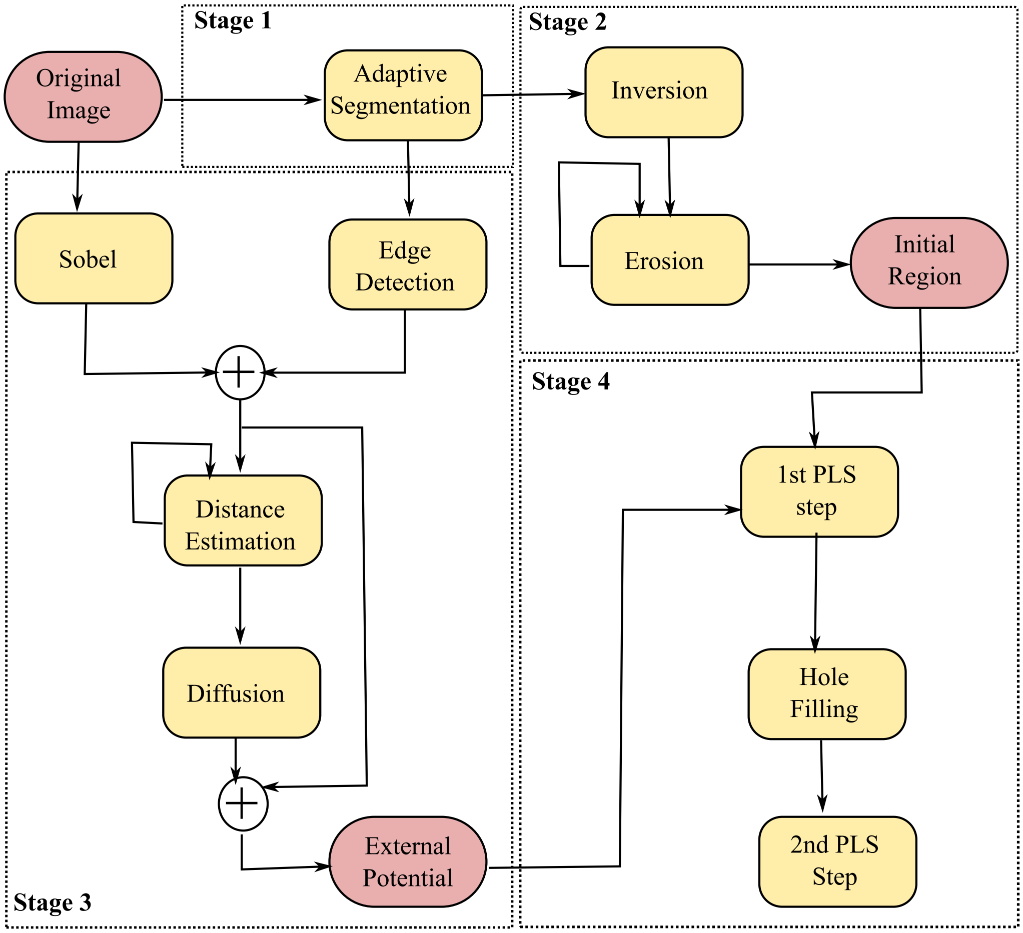

Fig. 3.7: Flow diagram of the latest algorithm version using region-based PLS. All the steps are implementable in a SIMD processor array

3.2.1 Stage 1. Vessel Region Pre-estimation

Due to the non uniformity of the gray level values along the vessels, an adaptive segmentation is needed. Instead of the complex CNN-based approach previously implemented, another strategy suitable for a hardware implementation has been employed (see Fig. 3.8).

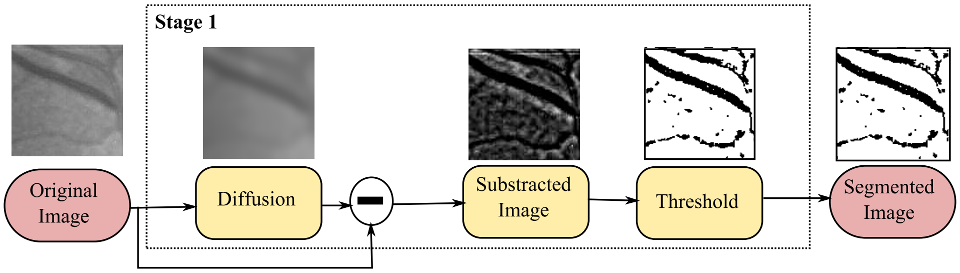

Fig. 3.8: Stage 1. Vessel region pre-estimation

Firstly, the original image is blurred by means of a diffusion step. This blurred image is used to estimate a local threshold value which 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 binarised based on a fixed threshold value. Notice that the equalisation step is not needed as well as the opening step proposed in the original version. This relies on the fact that the equalisation not only enhances the vessels but also it emphasises the noise contrast. So, the segmentation output using the equalisation resulted in a more noisy output. The opening step is not actually needed in this new proposal due to good results obtained.

3.2.2 Stage 2. Initial region estimation

Since a region-based PLS implementation is used in this case, the initial conditions for the PLS are regions instead of contours. The goal is still to assure that the initial region is completely outside of the vessel locations. So, the segmented image obtained in the *Stage 1* is inverted and eroded several times. The image obtained will be used as the initial conditions of the PLS, see Fig. 3.9.

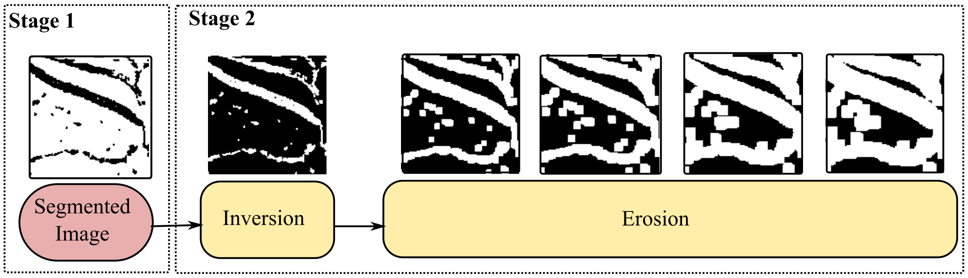

Fig. 3.9: Stage 2: Initial region estimation

3.2.3 Stage 3. External Potential Estimation

The main issue of this stage is to obtain a external image with an appropriate potential field to guide the PLS evolution, and sharp and emphasised locations in the edges of the vessels as the main criteria to stop the evolution, since this PLS version flows through homogeneous areas independently of the potential value. The processing steps performed in this stage are illustrated in Fig. 3.10.

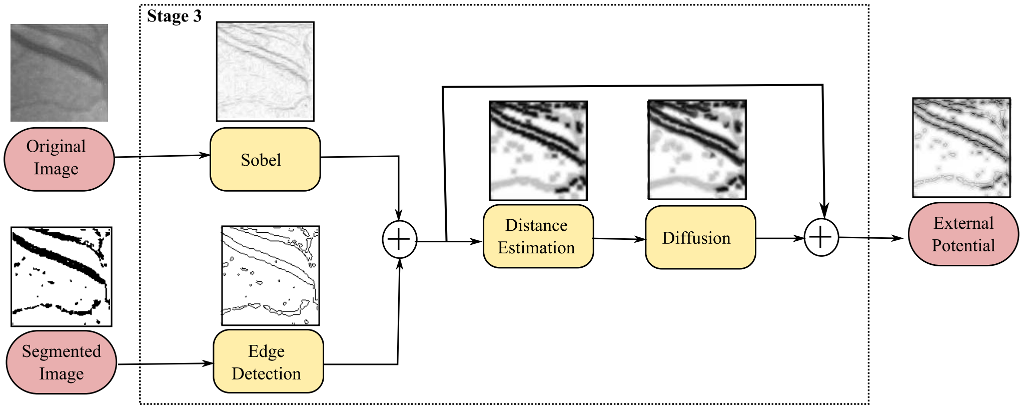

Fig. 3.10: Stage 3: External potential estimation

Firstly, 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. A distance estimation to the vessel locations is performed by means of several applications of a dilation operation, see Fig. 3.11. This creates the potential field needed by PLS to flow through. PLS will evolve guided by a stronger external potential in areas close to the edge, and vice versa. A diffusion step is made to alleviate the complexity of the directional gradient estimator in the GFE module. Finally, since the diffusion also smoothes the edges, the image containing the edges is added again in order to define a sharp potential valley to stop PLS evolution.

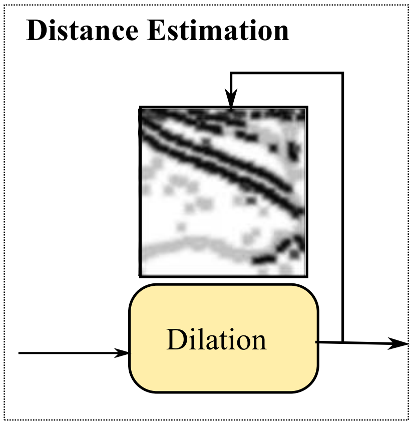

Fig. 3.11: Several dilations are made for the distance estimation function

3.2.4 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 in this stage is to determine the parameters which control the evolution. The external potential guides the PLS evolution towards the vessel edges, whereas the internal potential avoids 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 Fig. 3.12): a first PLS step, then a hole filling operation and finally, a 2nd PLS step.

During the 1st PLS step, the external potential is too weak to move the regions, because these are located far from the vessel edges. So, 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, it

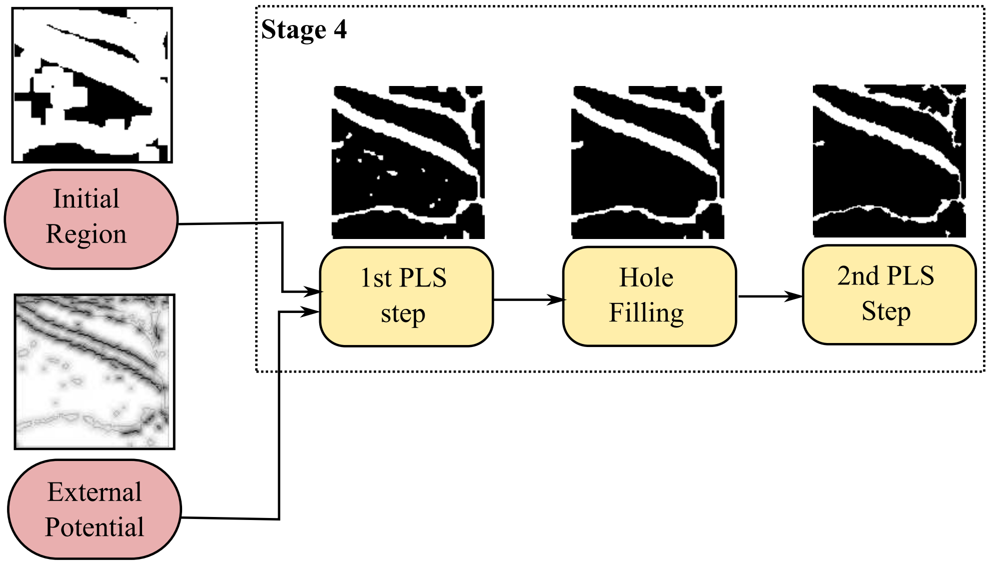

Fig. 3.12: Stage 4: PLS evolution. Notice that these steps have been proposed for a better control of the evolution towards the vessel edges

can be assured that the region merging will not affect to 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 Fig. 3.13). Notice that the hole filling operation is applied only once, in contrast with the original approach where the hole filling operation is made in each cycle of the PLS evolution. Finally, during the 2nd 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 should be weaker than the external potential, since PLS regions are supposed to be closer to the vessels. In this case, external potential should guide the evolution in order to provide a more accurate fitting of the exterior of the vessels. In this step, region merging is prevented in order to maintain the vessel topology. Furthermore, the internal potential has a higher influence to prevent the evolution through small cavities or discontinuities in the vessel topology.

Since only expansions are actually made, PLS performance can be optimised. In this sense, the eight iterations needed for a cycle can be reduced to four since compression forces are not needed.

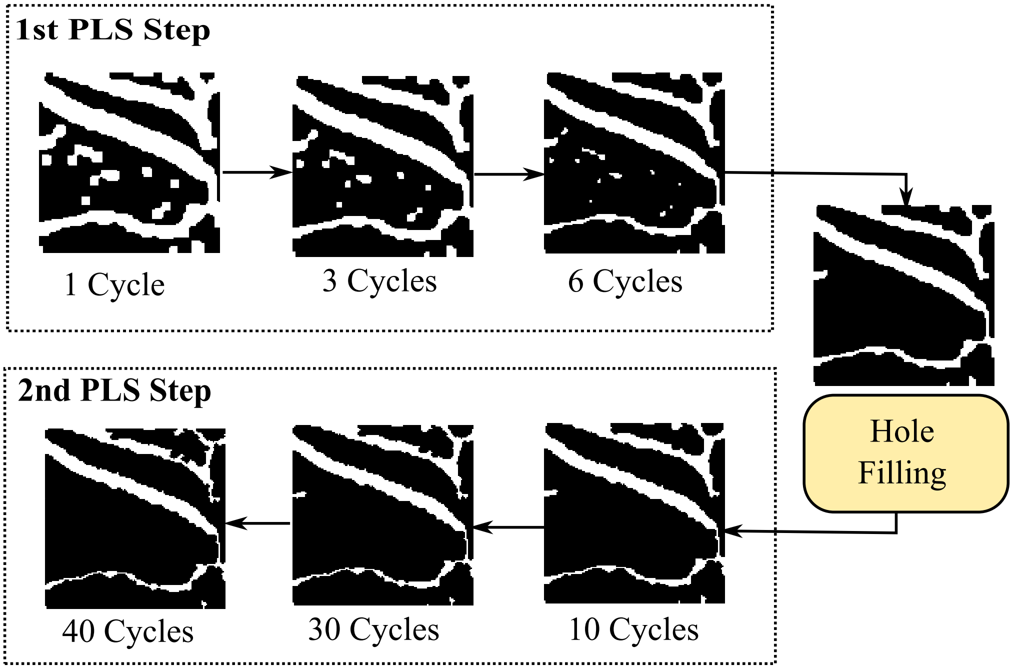

Fig. 3.13: Stage 4: Detailed step evolution. Notice that the hole filling operation is proposed to remove noisy regions and points inside the vessel structures

3.3 Summary

In this chapter, the automatic retinal vessel tree extraction algorithm has been presented and detailed. The first attempt consisted on defining a strategy to deal with the task in an automatic way, trying to fit the vessels from an image processing point of view. The main input images needed by PLS are automatically estimated from the original retinal image. Then, PLS were calibrated to tune the performance of the algorithm in order to achieve the vessel locations. The original version of the algorithm was implemented using CNN-based operations. Although the analysis of the results shows that the algorithm fit the vessels, some of the proposed steps, particularly the adaptive segmentation, cannot be implemented in the current processor arrays. So, a modification of some steps have been proposed in the final approach regarding the implementation in a pixel parallel processor array. This last approach is fully implementable in a SIMD chip. The region-based PLS version has been introduced since they simplify the topological transformations as well as alleviate the computation effort required respect with the previous contour version. In contrast, in the final approach of the algorithm only one hole filling is actually made to remove noisy structures. The computation of the external potential estimation has been also revised to get a better control over the evolution.

A qualitative comparison between both retinal vessel tree extraction algorithm approaches is presented in Table. 3.1.

	Original approach	**Final approach**
Computing platform	CNN	SIMD
HW implementation	Partial	Complete
Contour representation	Contour	Region
Design	General purpose	Specific purpose
Execution time of a PLS cycle[1]	518 μ s.	273 μ s.

Table 3.1: Comparison of the main features of both versions of the retinal vessel tree extraction algorithm

Notice that the final approach can be implemented in a SIMD processor array, and PLS can be customised for the particular task of retinal vessel tree extraction. So, the execution time of a cycle can be significantly reduced.

[1] Data obtained from the execution in the SCAMP-3 vision system [Vilariño and Dudek, 2007]. Notice that the execution time shown for the original approach is the optimised PLS version without the hole filling operation

Chapter 4

Experimental results: reliability and time performance

The retinal vessel tree extraction algorithm proposed in this thesis has been analysed from two points of view: the reliability and the time performance. The publicly available retinal image database DRIVE [Drive, 2008] has been used to test the reliability of the obtained results. The execution time has been analysed based on the implementation made on the SCAMP-3 vision system [Dudek, 2006]. Due to the fact that the maximum size allowed in this chip implementation is 128x128 pixels, the high resolution retinal images are split up into 128x128 sub-windows (see Fig. 4.1). Notice that the sub windows from the limits of the retinal image are not considered for their processing. Every step of the algorithm has been individually applied to each of the sub windows. The final result is obtained by means of the union of all the sub windows.

Finally, sub window overlapping has been also studied in order to analyse the improvement of the accuracy and the influence on the execution time in this algorithm.

In the following sections, the reliability analysis is firstly described. Then the implementation in the SCAMP-3 is shown as well as a time analysis of all the stages and the global time required for the processing of a whole retinal image. Finally, the overlapping is presented and tested.

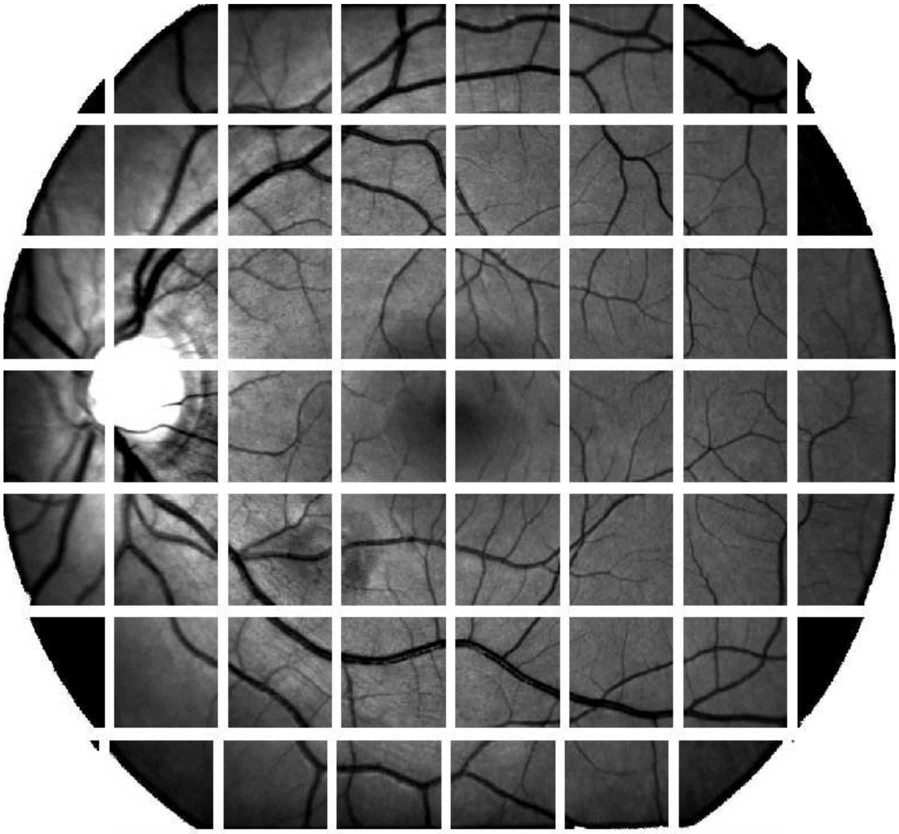

Fig. 4.1: 128x128 windows obtained from a retinal image

4.1 Reliability analysis

4.1.1 The DRIVE database

At the best of our knowledge, two publicly available data sets with retinal images (with and without diseases) can be found: STARE [STARE, 2008] and DRIVE [Drive, 2008] databases. Due to the content and type of images in these databases, their use is subject to the kind of application under study.

STARE (STructured Analysis of the REtina database) includes 20 images with a wide range and large-scalable pathological structures. One of the drawbacks of this database is the fact that there are a substantial inter-observer variabilities on the manual segmentations. Compare the amount of small segmented vessels between the two observers in Fig. 4.2.

The DRIVE database (Digital Retinal Images for Vessel Extraction data-

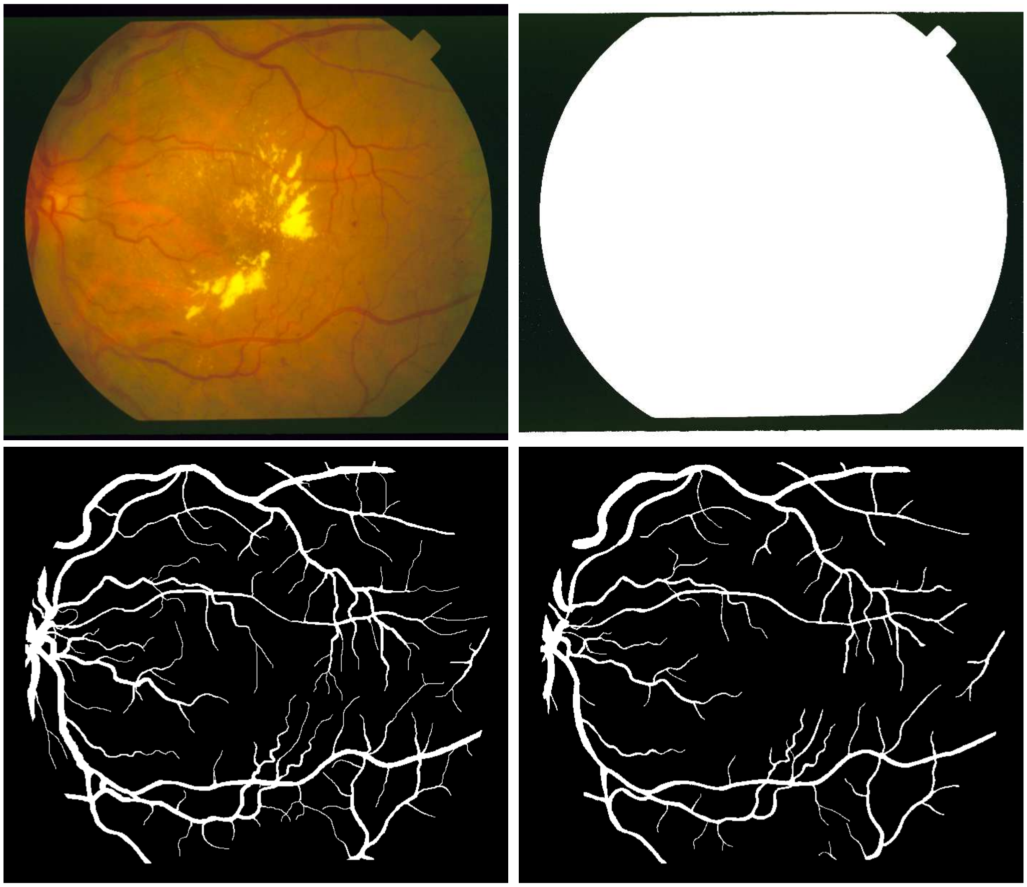

Fig. 4.2: Image from the STARE database.*1st row* Original image and mask delimiting the FOV. *2nd row* Manual segmentation of the first and second observer (notice the high inter-observer variability)

base) contains 40 available images with and without diseases, which allows a more general analysis of the applications, algorithms or methodologies under study. The inter-observer agreement is better than in the STARE database, see Fig. 4.3. The main goal of the DRIVE database is to establish and facilitate comparative studies on segmentation of retinal blood vessels in medical images. In this thesis, the DRIVE database has been selected for analysing the reliability, since the domain of this database is the retinal vessel tree extraction.

The images of the DRIVE database were obtained from a diabetic retinopathy screening program in the Netherlands. The screening population consisted of 400 diabetic subjects between 25-90 years old. Forty of them have been randomly selected, 33 images do not show any sign of diabetic retinopathy and 7 show signs of mild early diabetic retinopathy. The images were acquired using a Canon CR5 non-mydiatric 3CCD camera with a 45 degree field of view (FOV). Each image was captured using 8 bits per colour plane at 768 by 584 pixels, and they have been cropped around the circular FOV, which has a diameter of approximately 540 pixels. A mask image delimiting the FOV of each image has been provided (see example in Fig. 4.3).

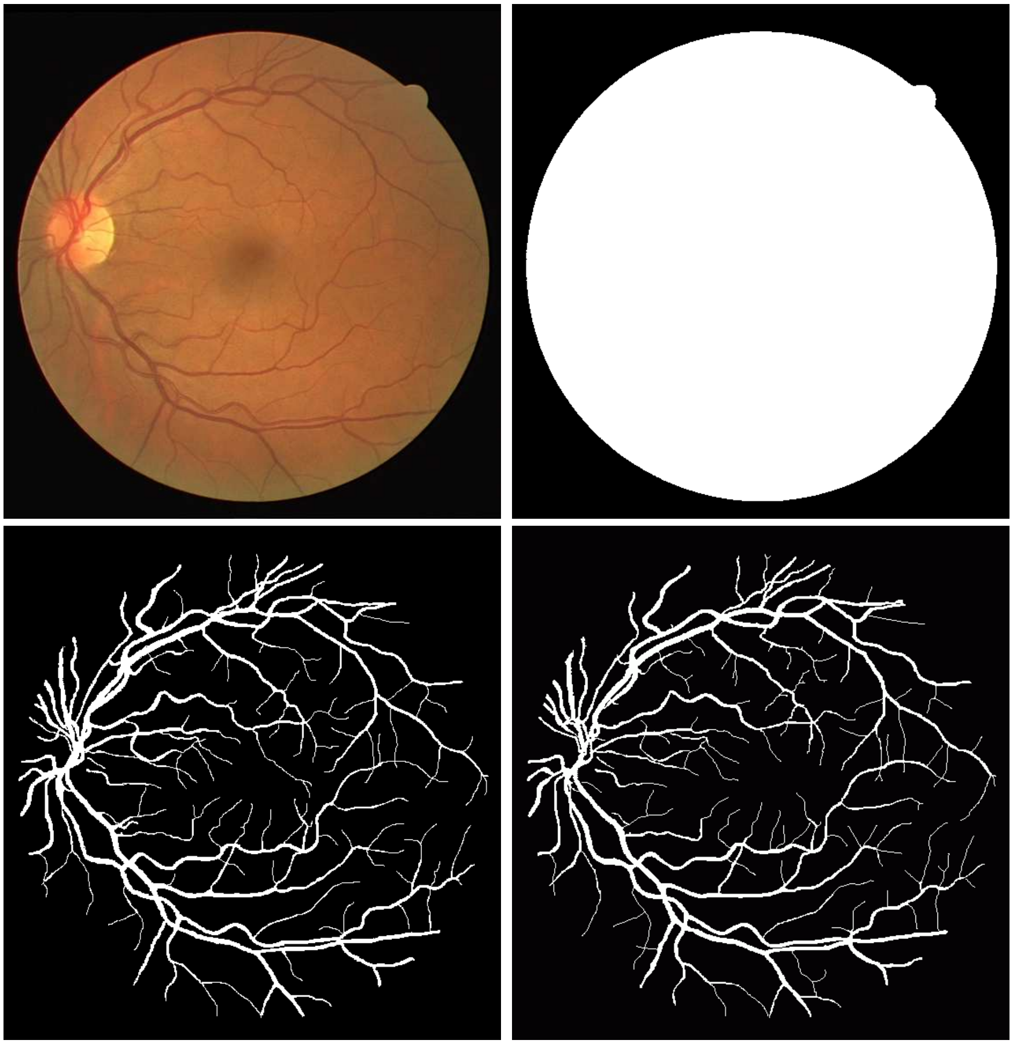

Fig. 4.3: Image from the DRIVE database.*1st row* Original image and mask delimiting the FOV. *2nd row* Manual segmentation of the first and second observer

The set of 40 images is divided into two sets, the training and the test sets, both containing 20 images. Three specialists (also called observers), trained by an experienced ophthalmologist, segmented several images. They were asked to mark all pixels for which they were for at least 70 % certain that they were vessel. The first observer segmented 14 images of the training set while the second observer segmented the other 6 images. The test set was segmented twice resulting in a set X and Y. Set X was segmented by both the first and second observer (13 and 7 images respectively) while set Y was completely segmented by the third observer. In set X about 12.7% of the pixels belongs to vessels. The difference between the first observer (gold standard[1]) and the second observer shows that vessel segmentation is not an easy task. Choosing a good criterion to measure the performance of vessel segmentation algorithms is not trivial, since it is highly dependent on the application in which the algorithm is used. The maximum average accuracy is usually used as criterion to see which method could most accurately segment the images in the database with respect to the gold standard. However, wider vessels have a larger influence on the final result than the smaller ones.

The test set is used to compute the accuracy in order to be compared with other proposals. The pixels are mainly classified into two classes, vessel and background. Only the pixels inside the FOV area, given by the provided FOV image mask, are taken into account to compute the accuracy. A comparison between the two manual segmentations has been also made in [Niemeijer et al., 2004] in order to show the difficulty of the segmentation task, even for human specialists.

4.1.2 Analysis of the reliability

The images of the DRIVE database have been used to test the accuracy of the results of the proposed retinal vessel tree extraction algorithm, in the MATLAB environment. These images have been split up into 128x128 sub windows. Each of the stages of the proposed algorithm have been individually applied to every sub window.

Initially, vessel regions are pre-estimated. Since the diffusion step achieves a good local threshold estimation, a threshold value of 5 has been established to refine the final result. Then, the initial region needed 4 erosion steps to assure that the initial regions are completely outside of the vessel locations. The external potential image was computed following the schema explained in the previous chapter. Finally, during the last stage the main parameters which controls the PLS evolution were tuned, see Table. 4.1. PLS evolves

[1] It is the observer taken as reference to compare and to test the results

during 6 cycles in the *1st PLS Step*. Notice that in this first step, the evolution is mainly controlled by the balloon potential since the initial regions are situated far from the vessel edges. Then, the hole filling operation is applied to remove noisy inner regions, and finally, in the *2nd PLS Step* evolve until the vessel edges are reached. In this case, the balloon potential influence is weaker due to the proximity to the vessel locations. The external potential will accurately guide the evolution, whereas the internal potential controls the smoothness of the shape.

	External Potential	**Internal Potential**	**Balloon Potential**	**Number of Cycles**
1st PLS step	100 %	1%	60%	6
2nd PLS step	100 %	30%	5%	*[1]

Table 4.1: Parameters established in the 4th Stage of the algorithm

The final result is formed by the union of all the sub windows. Figures from 4.4 to Fig. 4.12 shows the final result obtained by means of the algorithm proposed in this thesis.

[1] Notice that a convergence method has been implemented to fit the vessels, so PLS will employ the cycles needed for the processing of each image. It has been preestimated that a maximum number of 40 cycles is actually needed for an image.

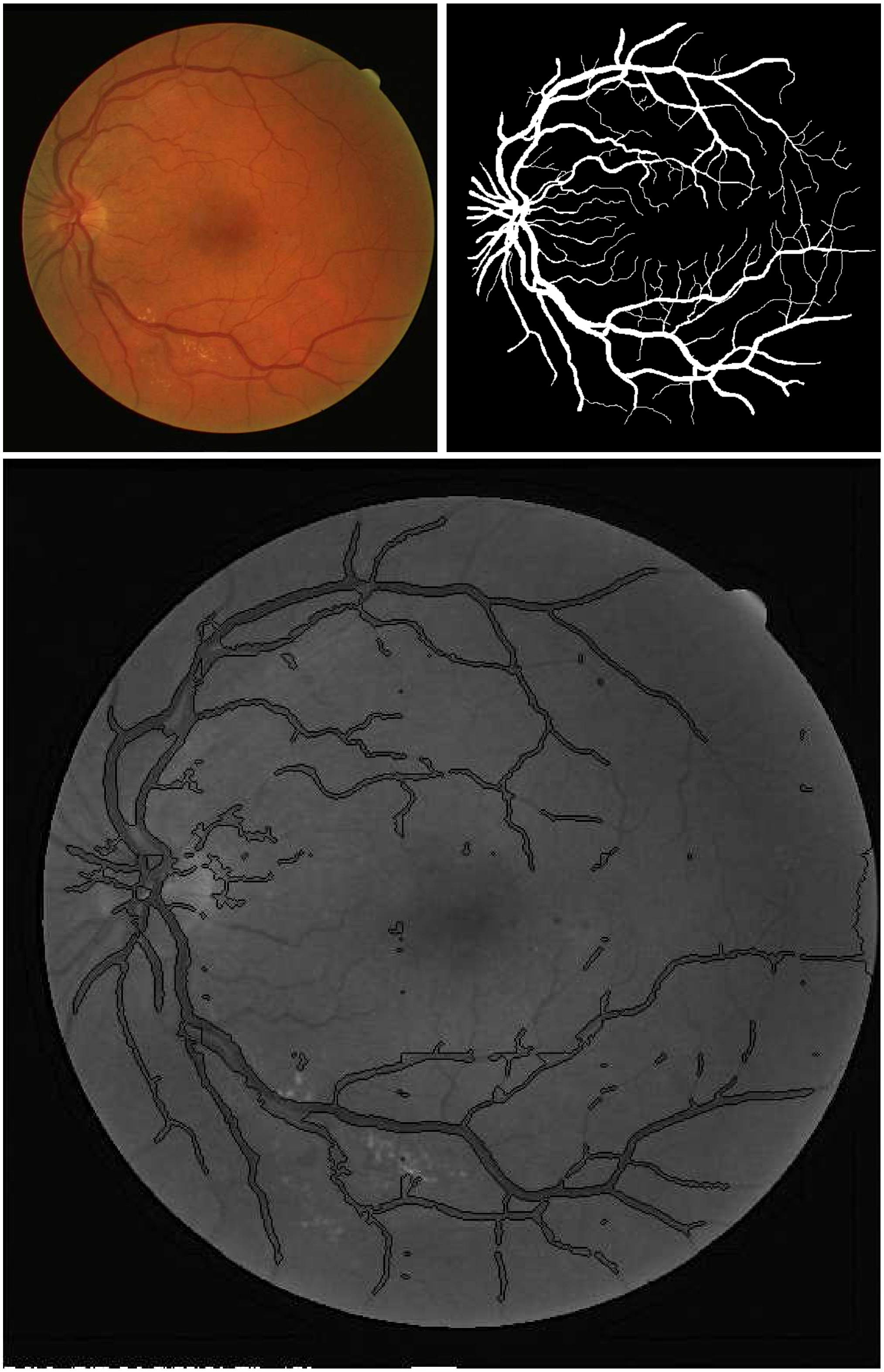

Fig. 4.4: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

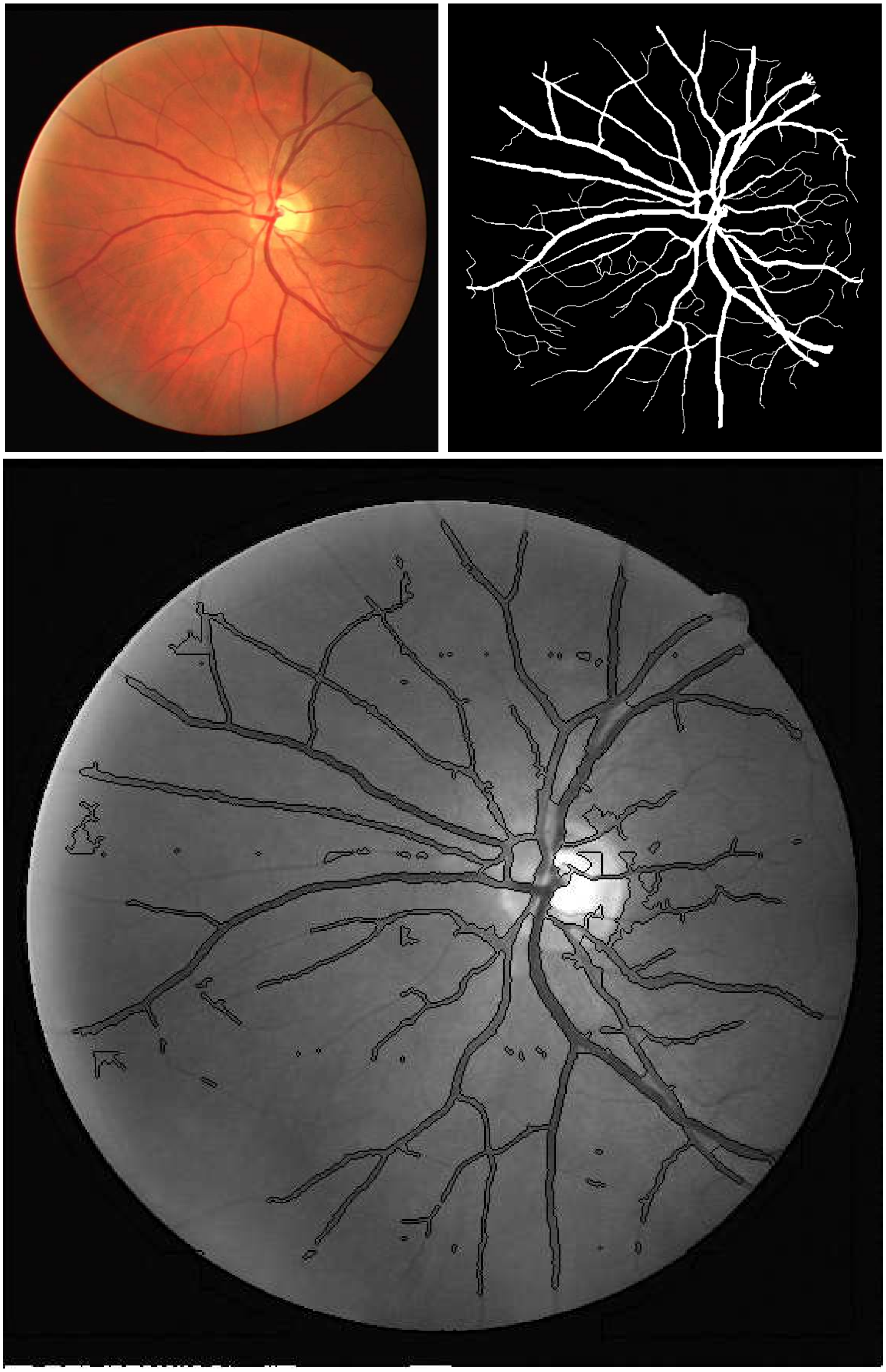

Fig. 4.5: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

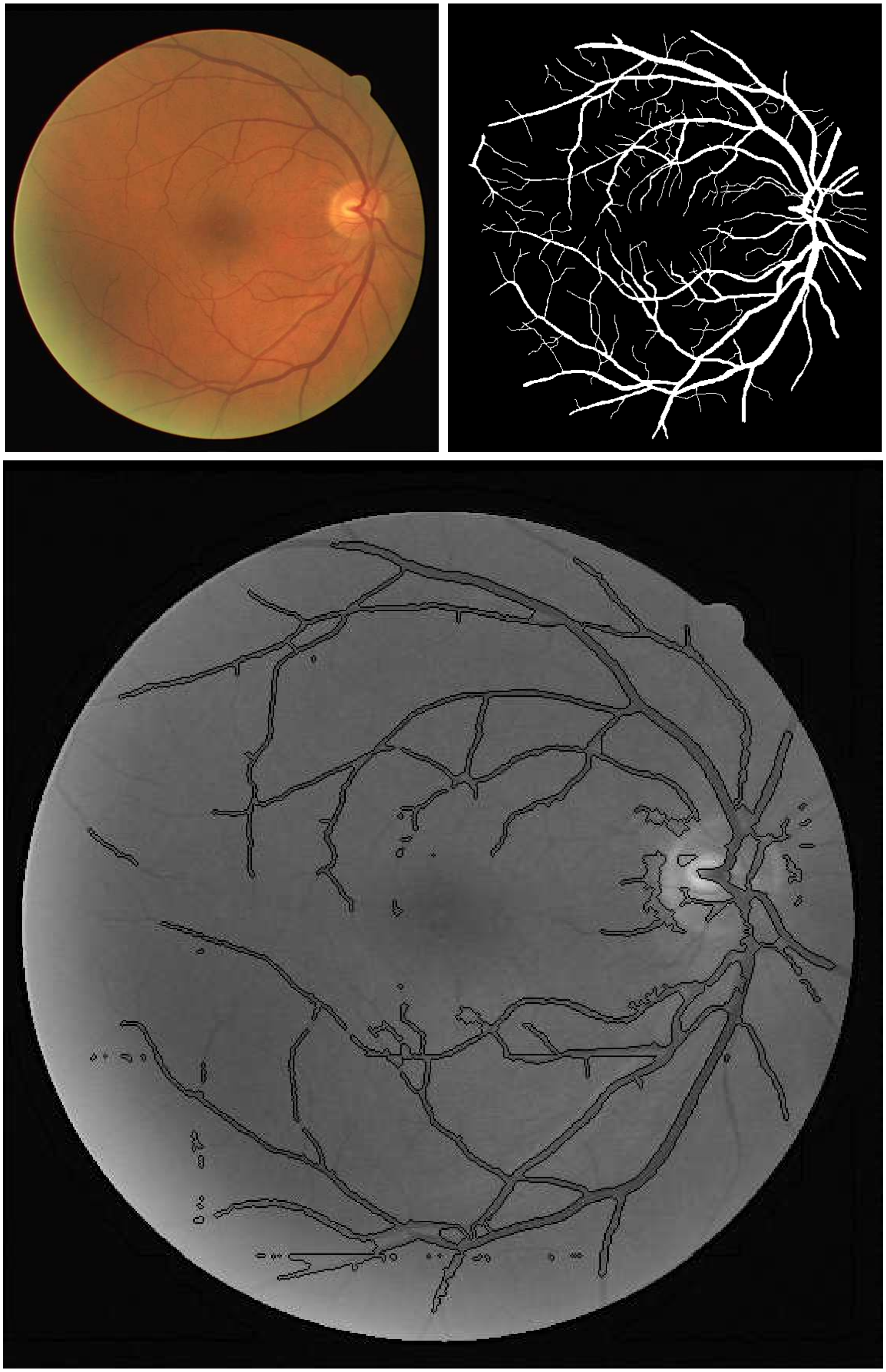

Fig. 4.6: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

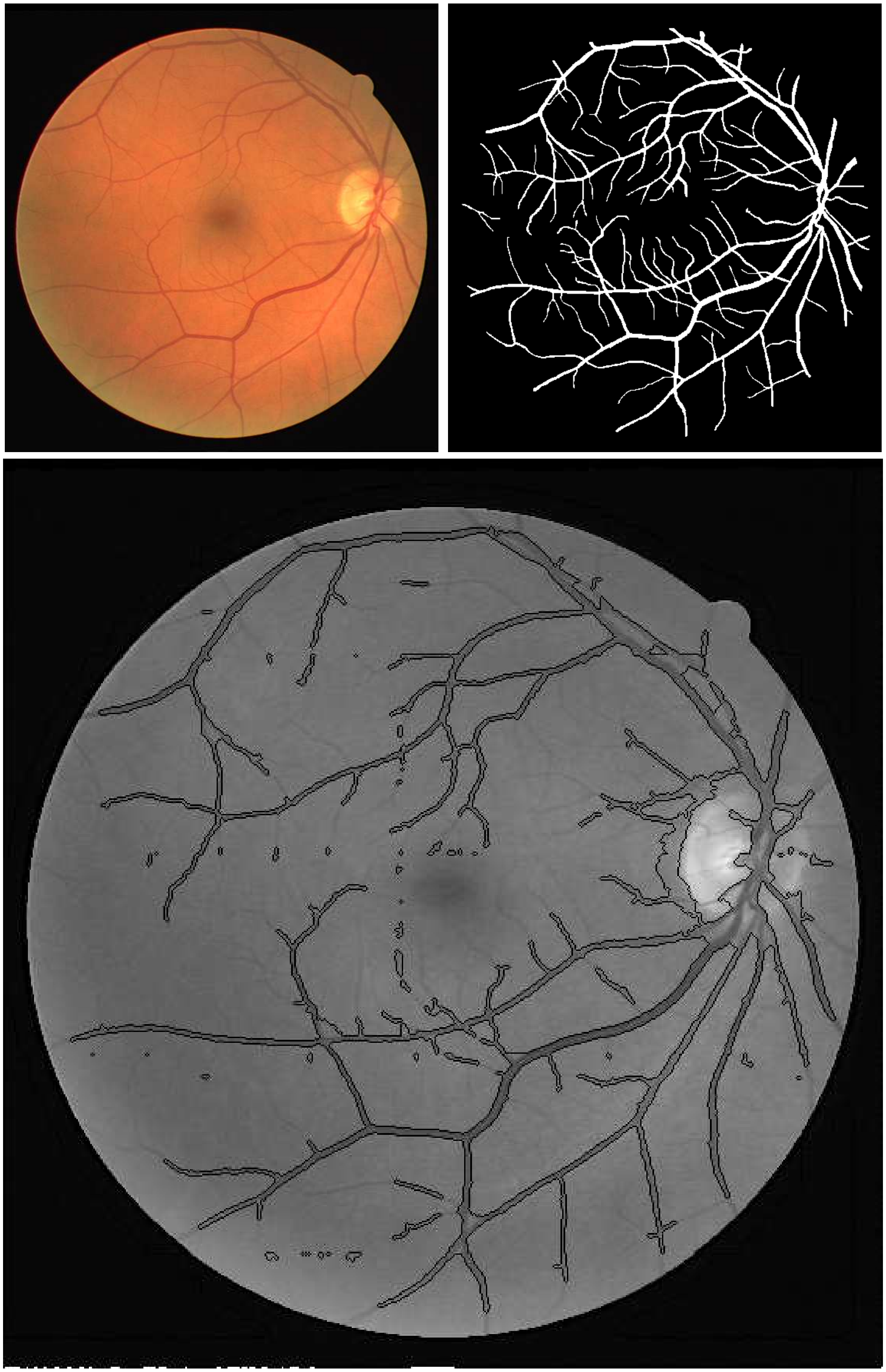

Fig. 4.7: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

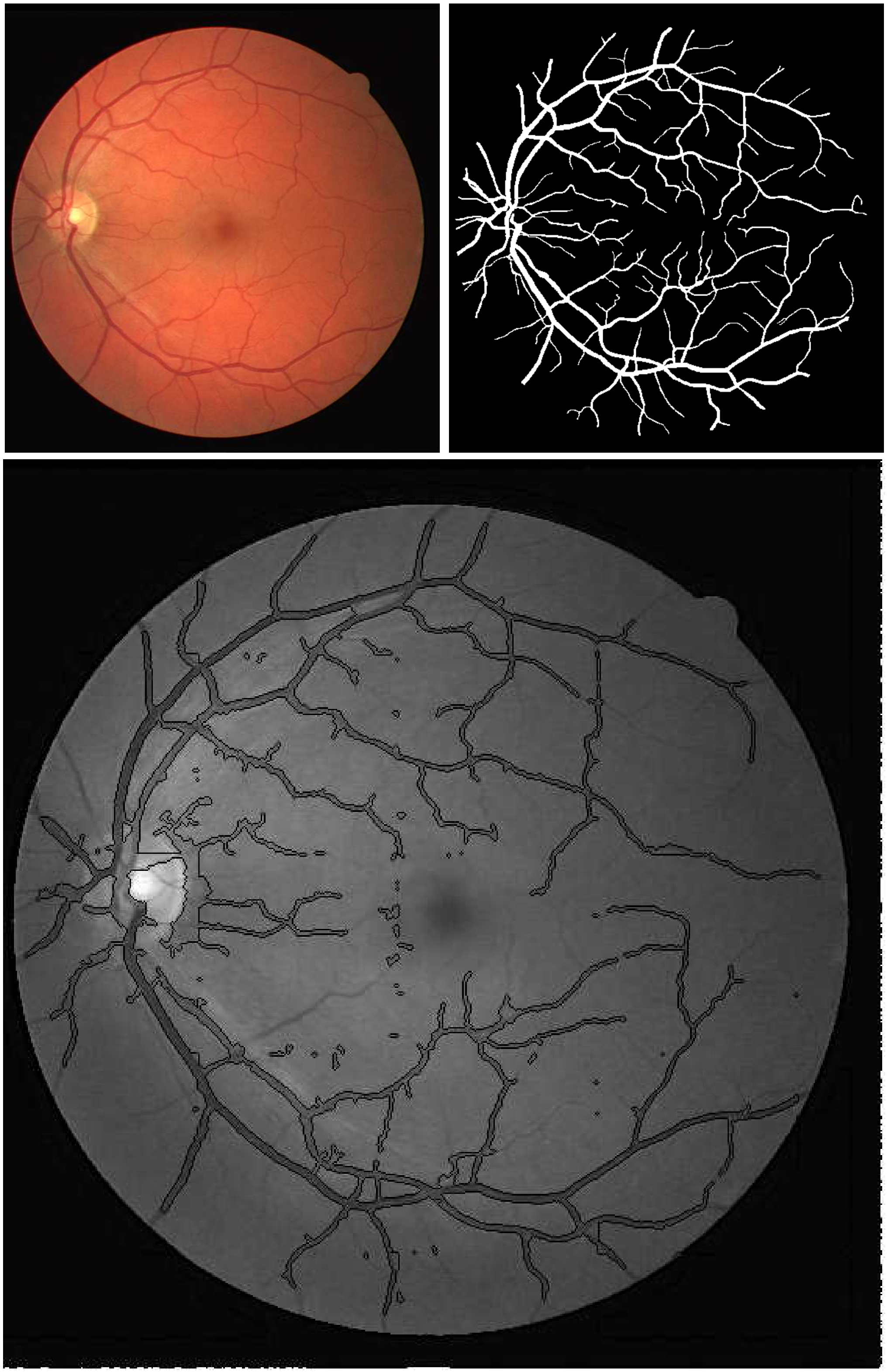

Fig. 4.8: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

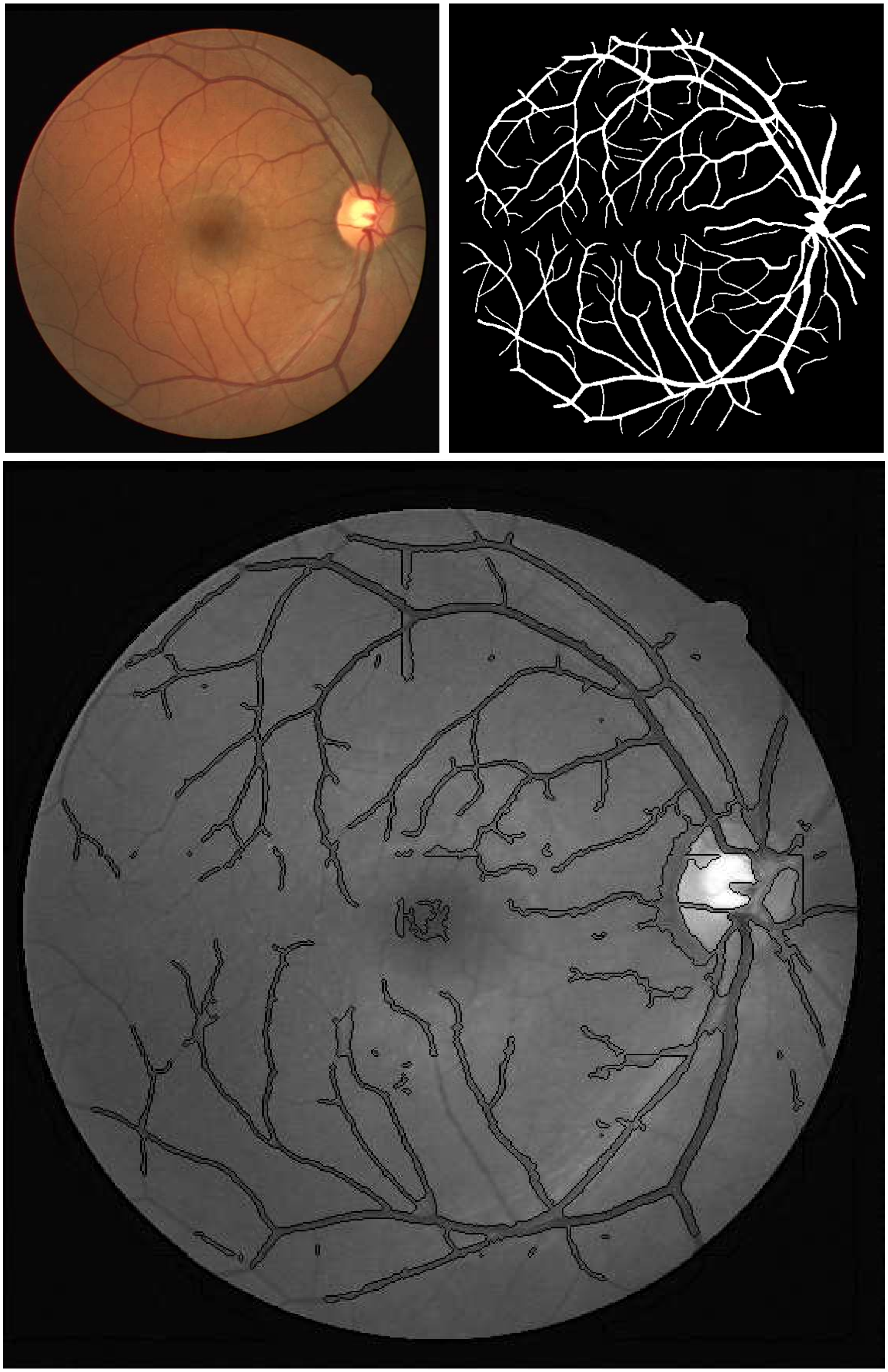

Fig. 4.9: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

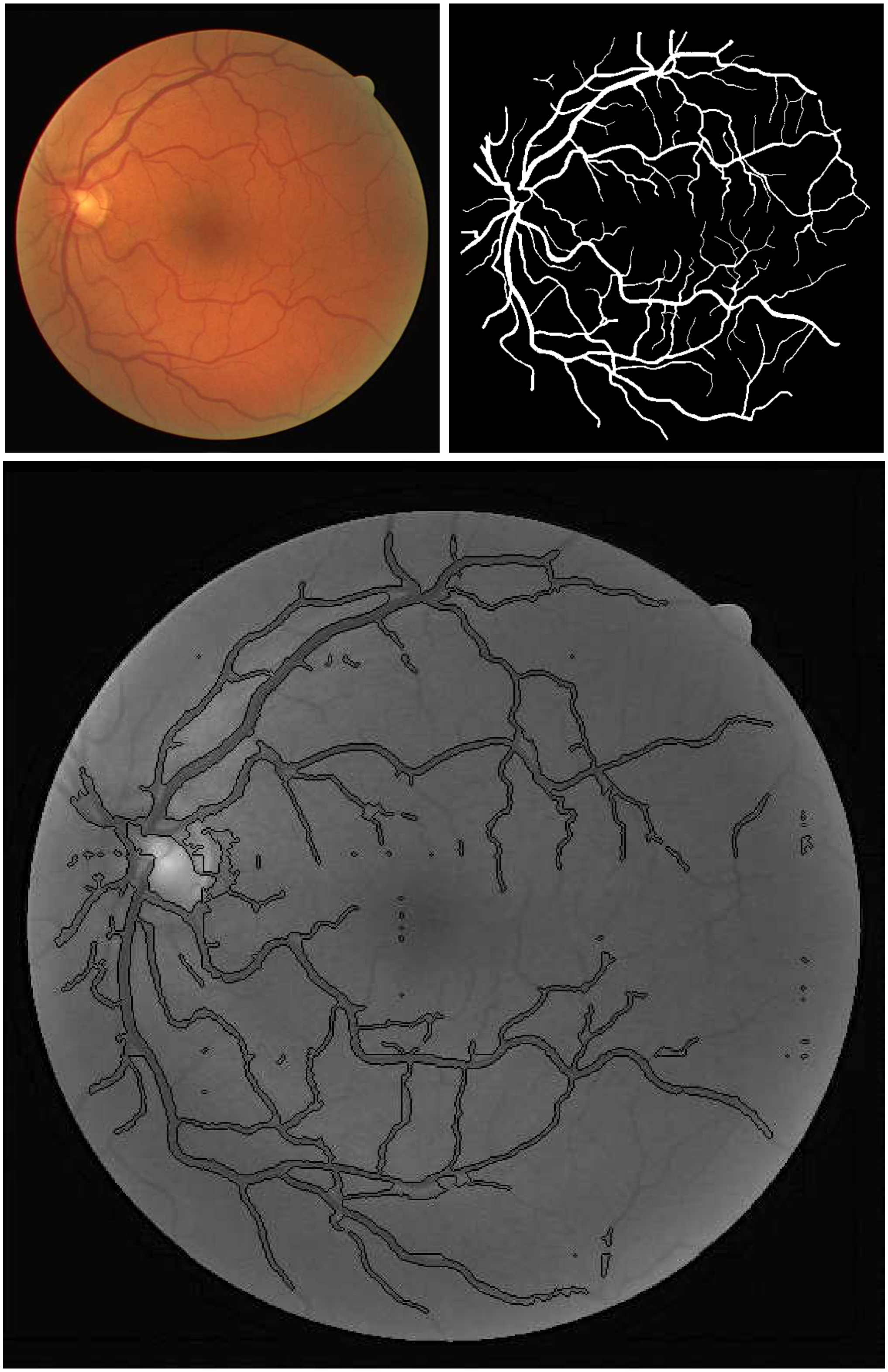

Fig. 4.10: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

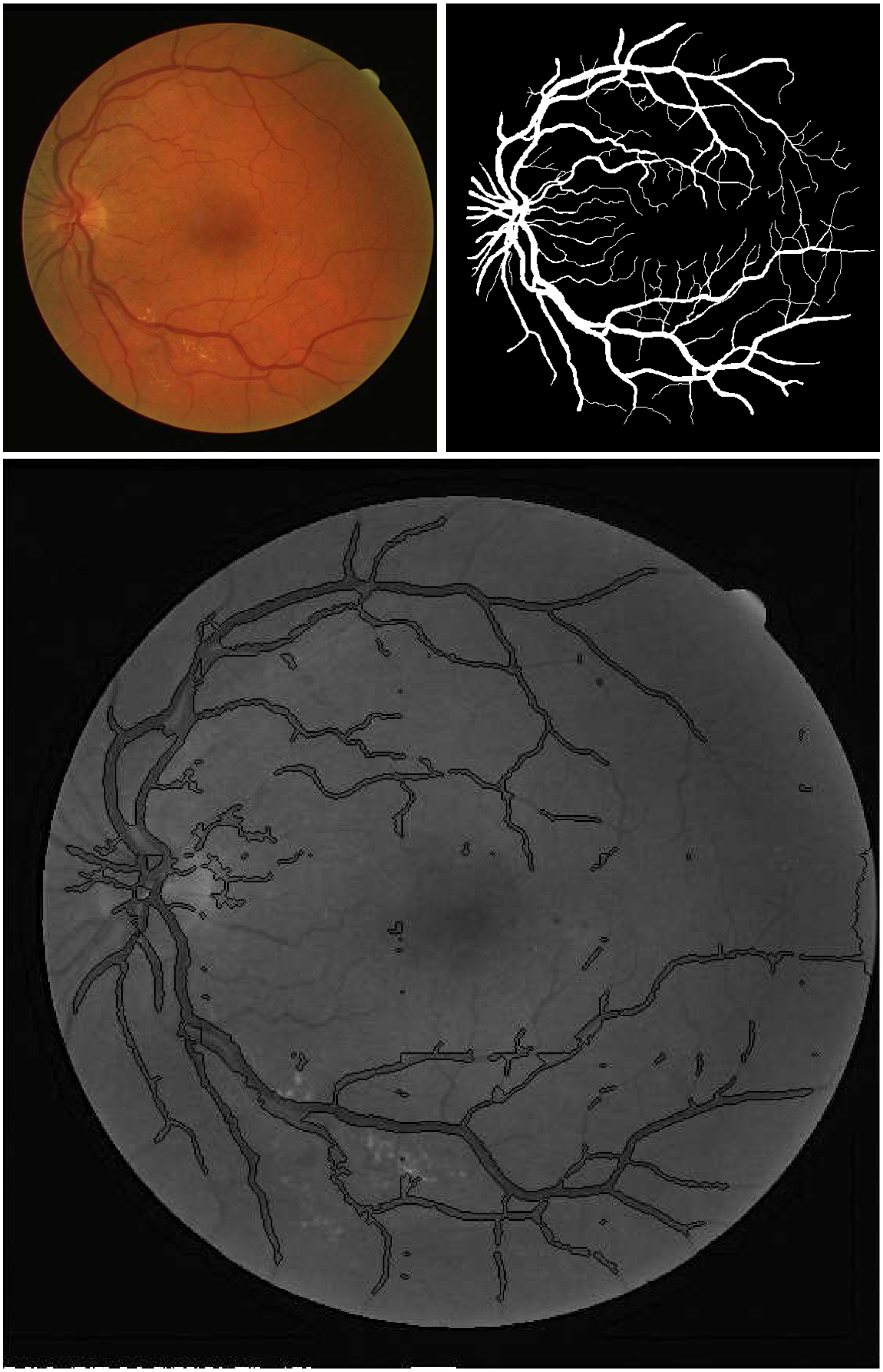

Fig. 4.11: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

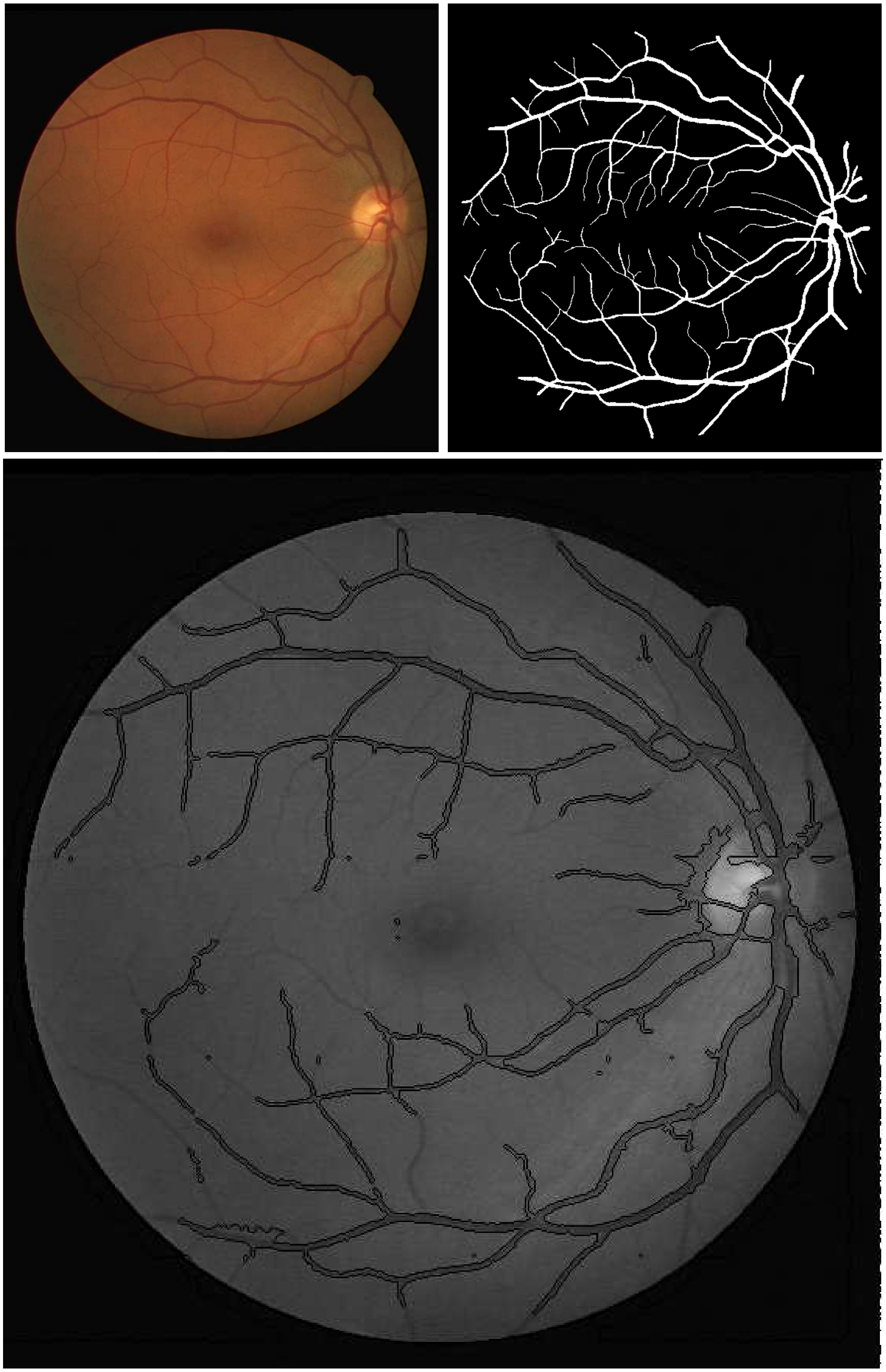

Fig. 4.12: *First row*: Original image and manual segmentation from the DRIVE. *Second row*: Final result obtained by the algorithm superimposed over the original image

A quantitative analysis is made to evaluate the quality of the obtained results. This analysis has been published in [Alonso-Montes et al., 2008c]. The accuracy (part of pixels correctly classified) has been presented in [Niemeijer et al., 2004] as a way to evaluate the reliability of the results in order to compare with other algorithms. The average value of the accuracy for all the images (the so-called maximum average accuracy, MAA) have been computed as follows:

$$Accuracy = \frac{T_{pos} + T_{neg}}{NP} \tag{4.1}$$

where T_{pos} and T_{neg} are the vessel (true positive) and non-vessel (true negative) pixels which are correctly classified, respectively, and NP is the number of pixels considered into the FOV region. The test set (a total number of 20 images) has been used to evaluate the accuracy of the algorithm. The manual segmentation of the second observer has been used as the gold standard, whereas the FOV image mask was used to compute only the pixels inside in those regions. The MAA value obtained (see Table 4.2) is 0.9180, which is a high enough value for most of the practical applications. Although this value is not the highest one compared with other PC-based algorithms found in the literature, the improvement in the computation time will be significantly more remarkable than with any other proposals (see the following section for further details).

Method	*MAA*
Manual Method	0.9473
[Soares et al., 2006]	0.9466
[Al-Rawi et al., 2007]	0.9458
[Kirsch, 1971]	0.9151
[Staal et al., 2004]	0.9611
[Chaudhuri et al., 1989]	0.8773
Proposed algorithm	0.9180

Table 4.2: Maximum average accuracy (MAA)

4.2 Time performance analysis: SCAMP implementation

The retinal vessel tree extraction algorithm proposed in this thesis meets the requirements to be computed by means of a SIMD processor array. In this case, the proposed algorithm has been implemented in the SCAMP-3 vision system, see [Alonso-Montes et al., 2007]. The execution time of its performance using this chip is analysed in this section.

4.2.1 SCAMP-3 vision system

The SCAMP-3 vision chip provides a high-performance low-power solution for computer vision applications, such as robotics, surveillance, industrial inspection, etc. The processor array operates in SIMD (Single Instruction Multiple Data) mode, since the processing elements simultaneously execute identical instructions on their local data. The processors can also exchange information with their nearest neighbours. The combination of analogue circuitry and digital architecture results in efficiency in terms of performance, cost and power dissipation.

The SCAMP-3 vision system executes a sequence of simple array instructions, like addition, inversion, one-neighbour access, operating in a pixel-parallel fashion on 128x128 arrays, at a rate of 1.25 MOPS per pixel. A development software has been provided in order to enable the development and evaluation of image processing algorithms in custom applications. The software includes Simulator Environment for custom applications.

4.2.2 Implementation on the SCAMP

The proposed algorithm has been implemented in the SCAMP-3 vision system adapting the steps to the specific performance and requirements of this chip, see scheme in Fig. 4.13. Since the SCAMP-3 captures the input images through an optical sensor and there is no possibility of loading the registers from PC, the main issue is the analysis of the execution time required for the performance of the algorithm. Initially, the algorithm has been implemented in the simulator provided with the SCAMP-3 vision system, establishing the error model of the actual hardware which allowed an accurate simulation. The algorithm has been also tested in the actual chip.

Initially, vessel regions are pre-estimated (see Fig.. 4.13) by means of the blurring of the original image, using a fast diffusion step, implemented in the SCAMP-3 via a resistive grid structure. This image contains a local threshold which properly segments weak vessels from the background. The blurred image is subtracted to the original image. A fixed threshold value (in this case 0) is defined to refine the segmentation result. A boundary segmentation

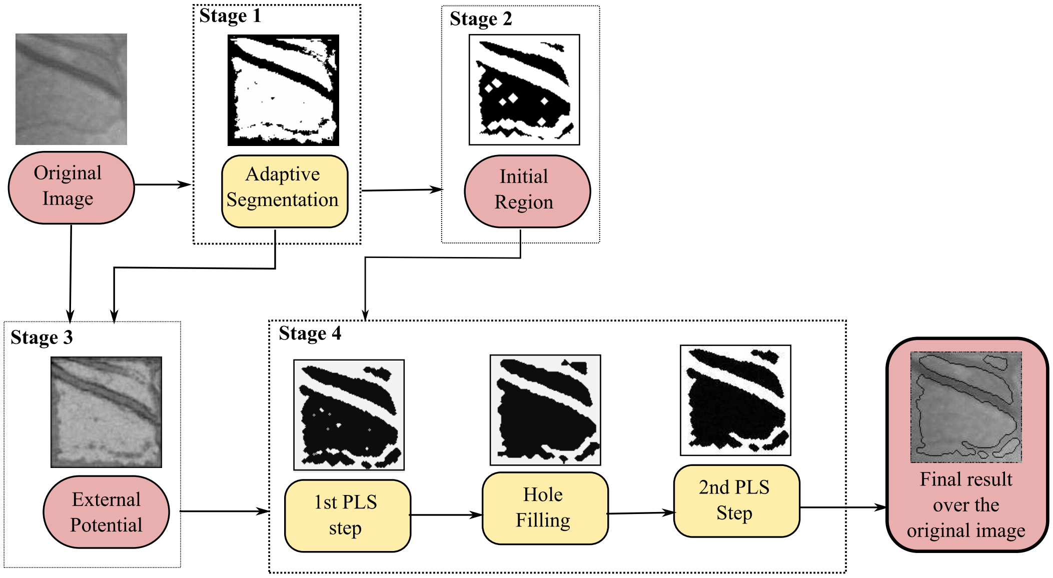

Fig. 4.13: Flow diagram of the implementation of the algorithm in the SCAMP-3 vision system

effect can be observed; this is due to a zero-padded boundaries in the diffusion operation. A post-processing technique, such as an overlapping, could solve this problem.

The initial regions, needed by PLS, are computed by means of eroding 4 times the inverted segmented image from the *Stage 1*.

The external potential estimation has been adapted to the specific requirements of the region-based PLS version implemented in the SCAMP-3 [Dudek and Vilariño, 2006b], see Fig. 4.15. Combining the Sobel result and the segmented image from the first stage, the vessel topology is maintained in those vessels with low contrast. The diffusion operation will create the potential

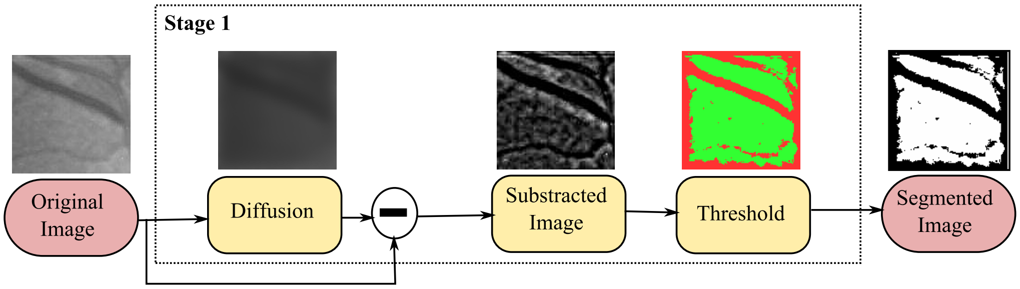

Fig. 4.14: Stage 1: SCAMP Adaptive segmentation

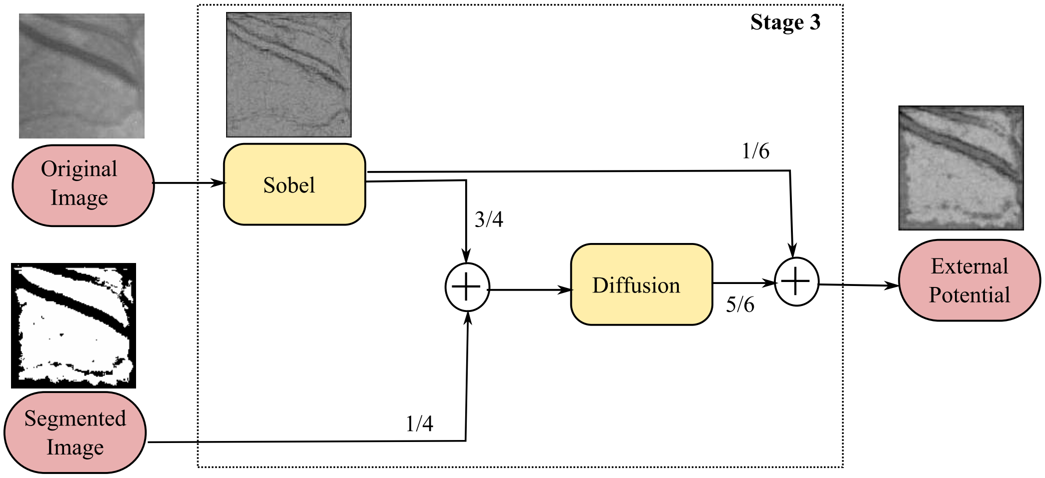

Fig. 4.15: Stage 3: SCAMP external potential estimation

field.

Using both the initial regions and the external potential images, PLS evolve to fit the actual vessel edges. In this stage, the main weighting values of the different potentials are tuned, see Table 4.3. During the 1st PLS step, a high balloon potential value is used during six cycles. This number of cycles has been selected based on the number of erosion steps applied on the Stage 2, which gives an approximation of the distance to the vessel edges. Since the adaptive segmentation gives only a pre-estimation of the vessel edges, this number of cycles could be increased to improve the PLS adaptation in the first PLS step. Then, the hole filling is applied to remove internal regions. Finally, the second PLS step is performed to fit completely the actual vessel edges. The number of cycles used in this step, has been empirically established at 40 cycles, since this number of cycles is sufficient in the evolution of PLS for all the sub-windows considered in this study. Notice that the number of cycles should be defined a priori. See the output

	External Potential	**Internal Potential**	**Balloon Potential**	**Number of Cycles**
1st PLS step	100 %	2%	32%	6
2nd PLS step	100 %	2%	4%	40

Table 4.3: SCAMP parameters used in the PLS evolution. Notice that the percentages are referred to the importance among the three potentials

of each stage in Fig. 4.16 and notice the adjustment of PLS to the vessel edges.

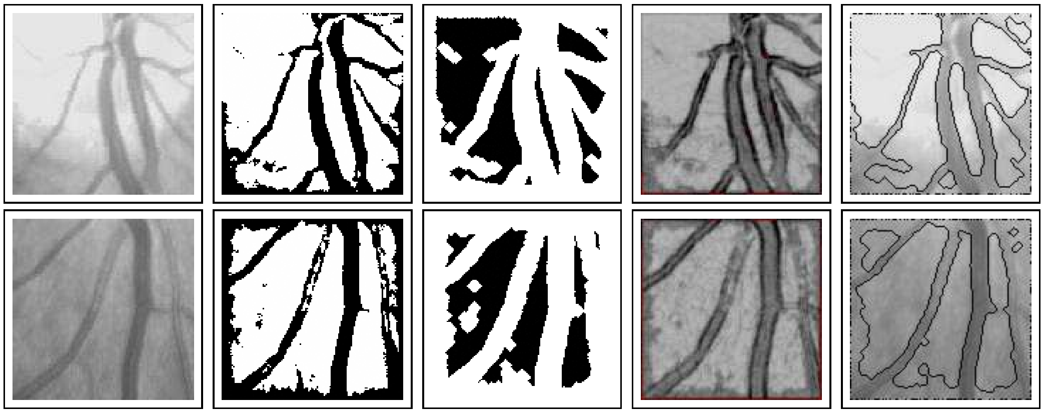

Fig. 4.16: Output images for each stage obtained with the SCAMP-3 *(from Left to Right)* Original image, vessel pre-estimation output, initial contours, external potential image, and final PLS output super imposed over the original image

4.2.3 Analysis of the execution time

The test of the computation speed on the current parallel processor, the SCAMP-3 vision system, has been analysed. The execution time required to perform the whole algorithm for a 128x128 sub-window in the SCAMP-3 vision system is about 6.5 ms (see the execution time required for each stage in Table 4.4). The total I/O time required for a 128x128 binary image is 1.25 ms [Barr et al., 2006]. Notice that the operation with a higher computation cost is the hole filling operation, since the 2nd PLS step consists on 40 cycles. The standard retinal image size of the DRIVE database is 768x584. So, a total number of 30 sub-windows is required. So, the global execution time required to process the whole retinal vessel tree is about 0.1925 s, excluding I/O operations. The execution time can be compared with the PC-based software implementations shown in Table 4.5. Comparing the execution time and the MAA previously obtained, this algorithm is fast maintaining a reasonable MAA value, compared with other proposals found in the literature, since conventional PC-based applications should get a compromise between speed and accuracy.

No. Stage	Stage	Exec. Time (μs)
1	Vessel Region Pre-estimation	12.8
2	Initial Region Estimation	55.2
3	External Potential Estimation	134.4
4	1st PLS Step (6 cycles)	518
	Hole Filling	1954.5
	2nd PLS Step (40 cycles)	3870.8

Table 4.4: SCAMP execution time for each stage, in a 128x128 sub-window

Method	*MAA*	*Exec. Time*
Manual Method	0.9473	2 h.
[Soares et al., 2006]	0.9466	3 min.
[Al-Rawi et al., 2007]	0.9458	5 s.
[Kirsch, 1971]	0.9151	2 s.
[Staal et al., 2004]	0.9611	15 min.
[Chaudhuri et al., 1989]	0.8773	5 s
Proposed algorithm	0.9180	0.1925 s.

Table 4.5: MAA and Execution Time

4.3 Remark: Overlapping

An analysis of the overlapping has been made to study the improvement in MAA value, since vessel continuity can be lost in the limits of the sub windows. This technique consists on overlapping rows and columns in the sub windows, in order to obtain redundant information in the limits areas (see Fig. 4.17).

The importance of the pixel information in the overlapping area depends on the position in that area and the distance to the limits of the image. In this sense, the pixel A in Fig. 4.17 has a higher value than A' in the image B. The main idea behind this process is the fact that the computation of a pixel on the limits of the image is less accurate than in the centre of the image. For example, in the overlapping area shown in Fig. 4.17, A and A' correspond to the same pixel, but its value is different since in image A is in next to the centre whereas in image B is next to the limit.

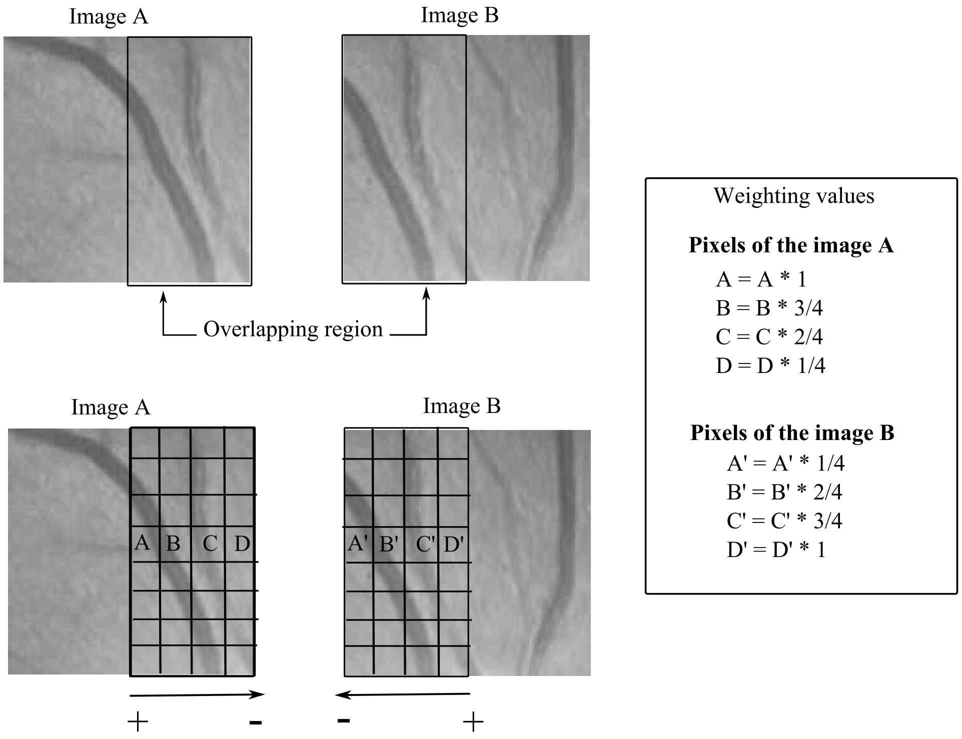

Fig. 4.17: Region of column overlapping and pixel value depending on their position and the sub window they belong to

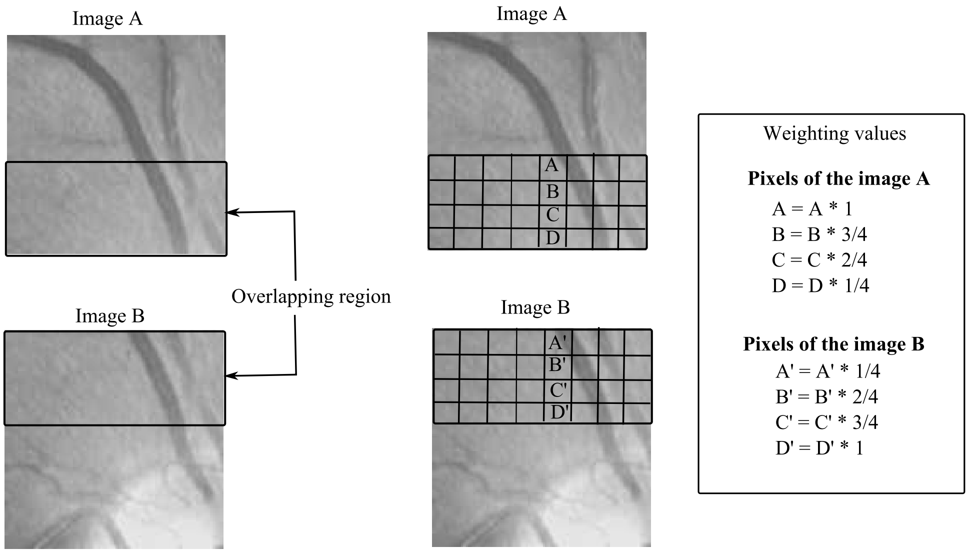

Fig. 4.18: Region of row overlapping and pixel value depending on their position and the sub window they belong to

The algorithm including the overlapping has been defined following the scheme showed in Fig. 4.19. The original image is split with redundant information on columns and rows. Each sub window is processed using the algorithm. Then, all the images with redundant information in the rows and columns are separately joined, in order to get the final result. The value in the redundant area is obtained by adding all of the values previously computed. Both images are added again and the result image is finally thresholded to get the final result.

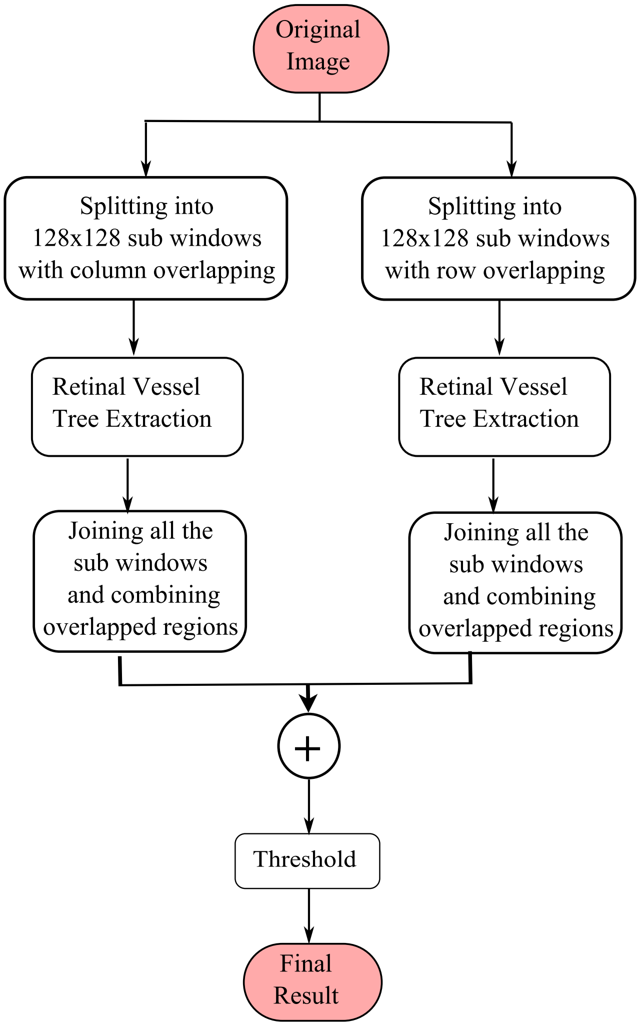
Original Image
Splitting into 128x128 sub windows with column overlapping
Splitting into 128x128 sub windows with row overlapping
Retinal Vessel Tree Extraction
Retinal Vessel Tree Extraction
Joining all the sub windows and combining overlapped regions
Joining all the sub windows and combining overlapped regions
+
Threshold
Final Result

Fig. 4.19: Flow diagram of the algorithm using the overlapping technique

Four main criteria have been studied in order to see the influence and to measure the improvement of the use of the overlapping technique: *accuracy, specificity, sensitivity and execution time.* Specificity and sensitivity are the true negative and the true positive ratios, respectively, expressed as percentages. Specificity are the true negative ratio, those background pixels which have been correctly classified, and sensitivity are the true positive ratio, those vessel pixels which have been correctly classified. Accuracy have been computed following the previously proposed equation (Eq. 4.1). According to this analysis, although the sensitivity value is improved with the overlapping size, specificity and accuracy have a slight improvement (see Fig. 4.20), and on the other hand, the required execution time increases significantly due to the increment of the sub window number which should be processed (see Table. 4.6 and Fig. 4.20).

Overlapping Size	**Number of Sub windows**	**MAA**	**Global Exec. Time (s.)**
0	30	0.9180	0.195
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

Table 4.6: Execution time and MAA depending on the execution time

In summary, it can be said that the overlapping technique is not actually needed due to the remarkable increment on the required execution time. Moreover, the specificity, sensitivity and accuracy values slightly improve using the overlapping.

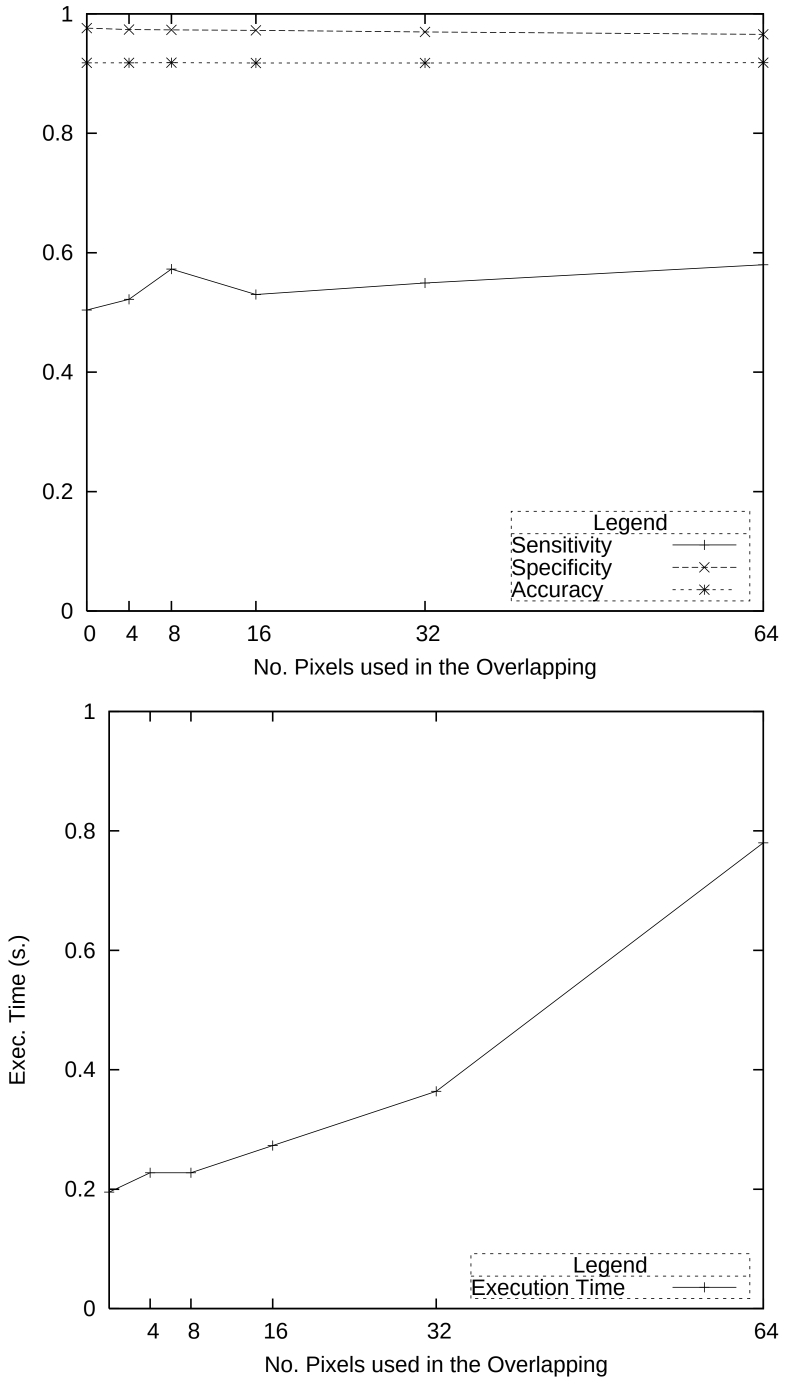

Fig. 4.20: Overlapping size influence in the specificity, sensitivity, MAA and execution time parameters

4.4 Discussion

The time performance as well as the accuracy of the results of retinal vessel tree extraction algorithm has been studied in this chapter. Initially, the accuracy has been tested using the publicly available database DRIVE in order to establish the quality of the results of the algorithm proposed in this thesis. This is a starting point to determine which kind of applications can use the results of our algorithm depending on the required MAA. The algorithm has been implemented in the SCAMP-3 vision system. The execution time obtained with this chip was analysed. Looking to the time performance needed for each stage, the hole filling is the operation with the highest computation effort. A total of 6.5 ms. is needed for the processing of a 128x128 sub window, whereas the whole angiography requires about 0.19 s. Although there are proposals in the literature that have a better MAA value, the execution time obtained for the algorithm proposed in this thesis is significantly lower (one or several orders of magnitude). Moreover, although computers are becoming faster, the window size of the current cellular processor arrays, like the Eye-RIS v.1.2 (QCIF) [Anafocus, 2008] is also increasing (176x144), which means that the number of needed sub windows will decrease as well as the execution time required for the task.

An overlapping technique has been also studied in order to analyse the improvement in the main criteria (MAA, specificity, sensitivity and execution time). Although there is a slightly improvement in the accuracy criteria, the execution time significantly increases so this overlapping was finally discarded.

Chapter 5

Applications

Retinal vessel tree is a key feature in a wide range of applications, from the medical domain up to the authentication applications tasks. The study and analysis of the vessel geometry features have become the basis of medical applications intended to be used in early diagnosis [Brieva et al., 2004], specially related with stenosis, malformations and cardiovascular risk, and effective monitoring of therapies in retinopathy [Miles and Nuttall, 1993]. Regarding retinal based authentication systems, the skeleton of the retinal vessel tree is considered one of the strongest biometric features due to the impossibility to forge it [Uludag et al., 2004].

In this chapter, the retinal vessel tree extraction algorithm has been integrated into several applications, particularly a personal authentication system and a medical application used to compute the arteriolar-to-venular ratio. This integration opens the future applicability of this proposal in a wide range of applications, specially regarding real-time requirement applications.

5.1 Retinal vessel authentication application[1]

Biometric features are inherently more reliable than passwords [Uludag et al., 2004], since they cannot be lost or forgotten, and they are extremely difficult to copy, share, distribute or forge. Among others, the retinal vessel tree is considered as a reliable feature. In the system addressed in [Mariño et al.,

[1] The authentication system proposed in this section is the basis of the thesis of Marcos Ortega Hortas from the Department of Computer Science, University of A Coruña.This was a collaborative work and it has been published in [Alonso-Montes et al., 2008a]

2006], this feature is used to authenticate individuals. The geometry of the retinal vessel tree, (the skeleton or crease), is extracted from a retinal image and registered. This skeleton is aligned with the one stored in the database, computing the cross-correlation index which determines the acceptance or rejection of the individual. The system proposed in [Ortega et al., 2006b] uses the feature points or landmarks of the retinal vessel tree to authenticate the individuals. Since the retinal vessel tree extraction implies a high computation effort, the use of the pixel parallel retinal vessel tree extraction algorithm is proposed, regarding the high computation speed. An analysis of the results is shown in order to see if the retinal vessel tree extracted using the algorithm proposed in this thesis is suitable for those systems. Since both applications use the skeleton of the retinal vessel tree, an skeletonisation step is introduced [Lam et al., 1992], see Fig. 5.1.

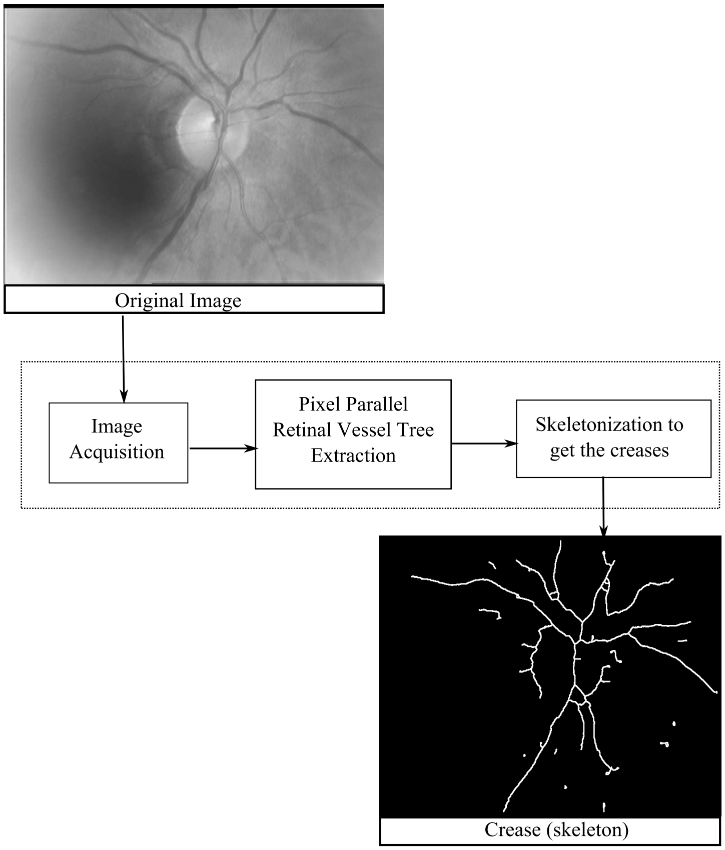
Original Image
Image Acquisition
Pixel Parallel Retinal Vessel Tree Extraction
Skeletonization to get the creases
Crease (skeleton)

Fig. 5.1: General scheme for obtaining the skeletons in order to be used in the authentication systems

5.1.1 Authentication system using creases

An authentication system based on the skeletons of the retinal vessel tree, also called creases, is proposed in [Mariño et al., 2006]. The flow diagram of the performance of this algorithm is shown in Fig. 5.2.

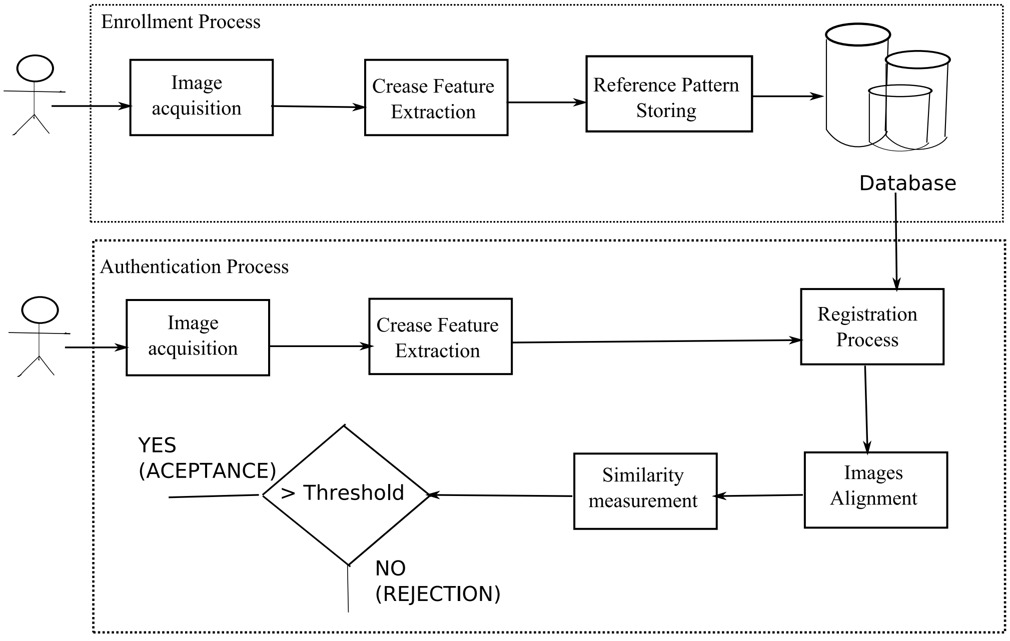

Fig. 5.2: Flow diagram of the authentication system addressed in [Mariño et al., 2006]

Initially, the extraction of a set of crease and valley lines is made to use as anatomical landmarks. This pattern is stored in a database of authorised people. When an individual needs to be authenticated, the acquired image is processed to extract the retinal vessel tree (crest and valley lines) and a registration process is required in order to align it with the pattern stored in the database. The registration process avoids miss-registrations between images due to inevitable eye movement. Once the images are aligned, a similarity measure is computed. A given threshold will determine the acceptance or rejection of the individual.

The retinal vessel extraction algorithm has been integrated into this application. The extracted retinal vessel tree is processed to get the skeleton. This skeleton, which belongs to the centre of the vessel is used as the crest and valley lines to authenticate the individuals.

Image registration

Once the creases have been extracted from the images, the registration process is performed. Then, an alignment of the images is obtained by matching sufficiently long portions of vessels. Notice that although the whole retinal is used, this methodology does not have a strong dependence on the quality of the segmentation. In this sense, the alignment must be robust to missing and non-overlapping features. The alignment function takes one of the images as the reference, while the other is iteratively transformed until the function reaches a global maximum. A suitable function to measure the quality of the alignment is the correlation function:

$$Corr_{\tau} = \sum_{x \in f} f(x) g(\tau(x)) \tag{5.1}$$

where f and g are the creaseness images and τ represents the transformation whose parameters need to be tested. These parameters as well as the function $Corr_{\tau}$ are not easy to optimise. The approach to overcome this fact is to search within the parameter space at multiple resolutions following the proposal in [den Elsen et al., 1995]. The ridgeness and valleyness images are at the bottom of the processing pyramid and each level is a sampled version of the previous at half resolution, until images have a final size of 64 pixels in each dimension. An exhaustive search starts at the top of the pyramid computing the correlation in the Fourier domain. Several values from each level are kept as seeds for the next level. This is made for both robustness of the methodology and improvement of the computation time. Then, the creases of the registered images are used to obtain a similarity measure between them. The normalised cross-correlation coefficient γ has been used as the similarity value, and it is computed as follows:

$$\gamma = \frac{\sum_{x,y} [f(x,y) - \overline{f}][g(x,y) - \overline{g}]}{\sqrt{[f(x,y) - \overline{f}]^2 [g(x,y) - \overline{g}]^2}} \tag{5.2}$$

where $\overline{g}$ is the mean of the registered image, and $\overline{f}$ is the mean of the image. Only the pixels belonging to the overlapping area are not null. Finally, a confidence value should be determined in order to distinguish if two images belongs to the same individual. Two images will be considered that they belong to the same person if the γ value is higher than this confidence value.

5.1.2 Authentication system using point feature extraction

The system addressed in [Ortega et al., 2006b] is based on the approach of the minutiae (ridge endings, bifurcations and endpoints) used in the fingerprint recognition. In this system, the landmarks (bifurcations) of the retinal vessel tree are extracted to characterise the individual. The use of this biometric feature points, instead of the whole retinal vessel tree, gives more robustness to the pattern, as well as an easier encoding and more efficient matching process. The proposed stages for this system are shown in Fig. 5.3. This work was presented in [Alonso-Montes et al., 2008a].

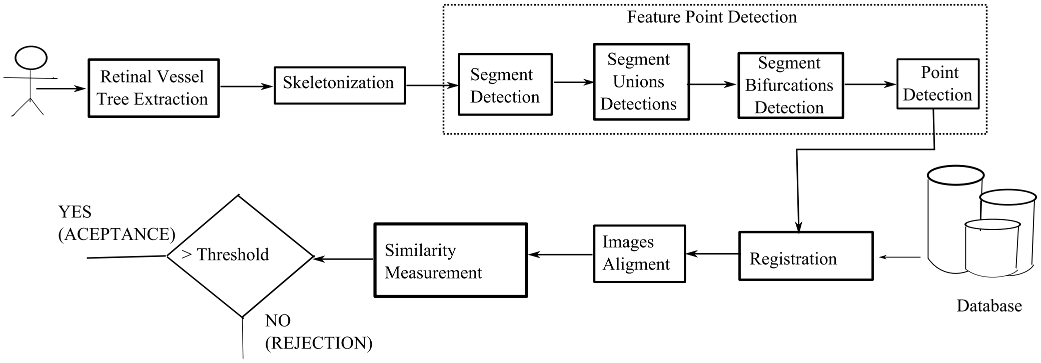

Fig. 5.3: Flow diagram of the authentication system addressed in [Ortega et al., 2006b]

Initially, all the authorised user vessel patterns are stored in a database. When an individual needs to be authenticated, a new retinal image is acquired. The feature points are extracted and a registration process is made to align those points with the pattern of the individual, which is stored in the database. Once the images are aligned, a similarity measure is computed, and if it is higher than a given threshold the individual is accepted, otherwise he is rejected.

Point feature extraction

Four steps have been proposed to implement the point feature extraction: firstly, a segment detection is made, then union and bifurcations are computed, and finally the feature points are determined. Initially, the whole vessel tree is tracked in order to label the segments and their ending points.

When a non-zero pixel is detected, its non-zero neighbours are tracked recursively following the line directions. Once all the segments are labelled, the relationships between them, such as the unions and bifurcations, should be detected. Two close segments have an union relationship between them if one is continuation of the other in the same vessel. The bifurcation relationship detection is made by extending every segment from its endpoints along a fixed length. If another segment is found in the surroundings, those segments are marked as a bifurcation. Finally, some redundant points are removed.

Registration and matching

The proposed matching algorithm is based on the proposal described in [Tico and Kuosmanen, 2003]. The algorithm receives both of the sets of feature points, one with the points previously computed in the candidate image, and another one from the image stored in the authorised user database (the reference image). A matching value is computed to obtain the similarity between both images, which will determine the acceptance or rejection of the individual.

Retinal images of the same individual obtained at different moments will probably have some deformations, due to the eye movements. In this sense, the Similarity Transformation (ST) has been proposed, since it deals properly with images where the rotation angle is moderate. The scaling is nearly constant for all the images, due to the proximity of the eye to the camera. An initial set of possible transformations is made following the approach described in [N. Ryan and de Chazal, 2004], selecting pairs of points in both images. The similarity value, S, for a pair of points is computed based on the distance between them:

$$S(A, B) = 1 - \frac{distance(A, B)}{D} \tag{5.3}$$

where A and B are the points, D is the maximum distance for those points to be considered as a possible match. If this distance is greater than D then $S = 0$. In some cases, two close points (C_1, C_2) in the candidate image, could have a good value of similarity respect to a point R of the reference image. So, the most suitable matching pair is identified by means of the probability of correspondence, which is defined by comparing the similarity value between those points and the rest of similarity values of each one of them:

$$P(A_i, B_j) \quad = \quad \frac{S(A_i, B_j)^2}{\left(\sum_{i\prime=1}^{M} S(A_{i'}, B_j) + \sum_{j'=1}^{N} S(A_i, B_{j'}) - S(A_i, B_j)\right)} \tag{5.4}$$

A MxN matrix, Q, is computed, such that the position (i, j) holds the value $P(A, B)$ for all the combinations. The best non-zero values in each row and column are selected. Finally, the matching value that measures the similarity between both points sets is computed as follows:

$$\frac{1}{\sqrt{MN}} \sum_{(i,j) \in Q} S(A_i, B_j) \tag{5.5}$$

where M and N are the number of points of both feature point sets. This value is compared with a given threshold, which will be establish the acceptance or the rejection of the individual.

5.1.3 Experimental Results

The images used in the experiment for both authentication systems have been acquired during a period of 15 months in the University Hospital of Santiago de Compostela (CHUS), using a Cannon CR6-45NM Non-Mydriatic Retinal Camera, with a resolution of 768x584 pixels.

Initially, the retinal vessel tree has been extracted following the algorithm proposed in this thesis. The last step consists on a skeletonisation step which will be the input to both authentication systems. The original retinal image has been split into 128x128 sub windows. The whole retinal vessel tree extracted by means of the algorithm consists on the union of all the processed sub windows. Since the image resolution is 768x584, a total number of 30 sub-windows is required. So, the global execution time required to process the whole retinal vessel tree (excluding I/O operations) is 0.1925 s.

In the first proposal addressed in [Mariño et al., 2006] the whole retinal vessel tree is used as the biometric pattern to compute the similarity value. In [Ortega et al., 2006a], the biometric feature over the retinal vessel tree are computed (see Fig. 5.4), and then the registration and matching processes were made to authenticate the individual. This work was published in [Alonso-Montes et al., 2008a].

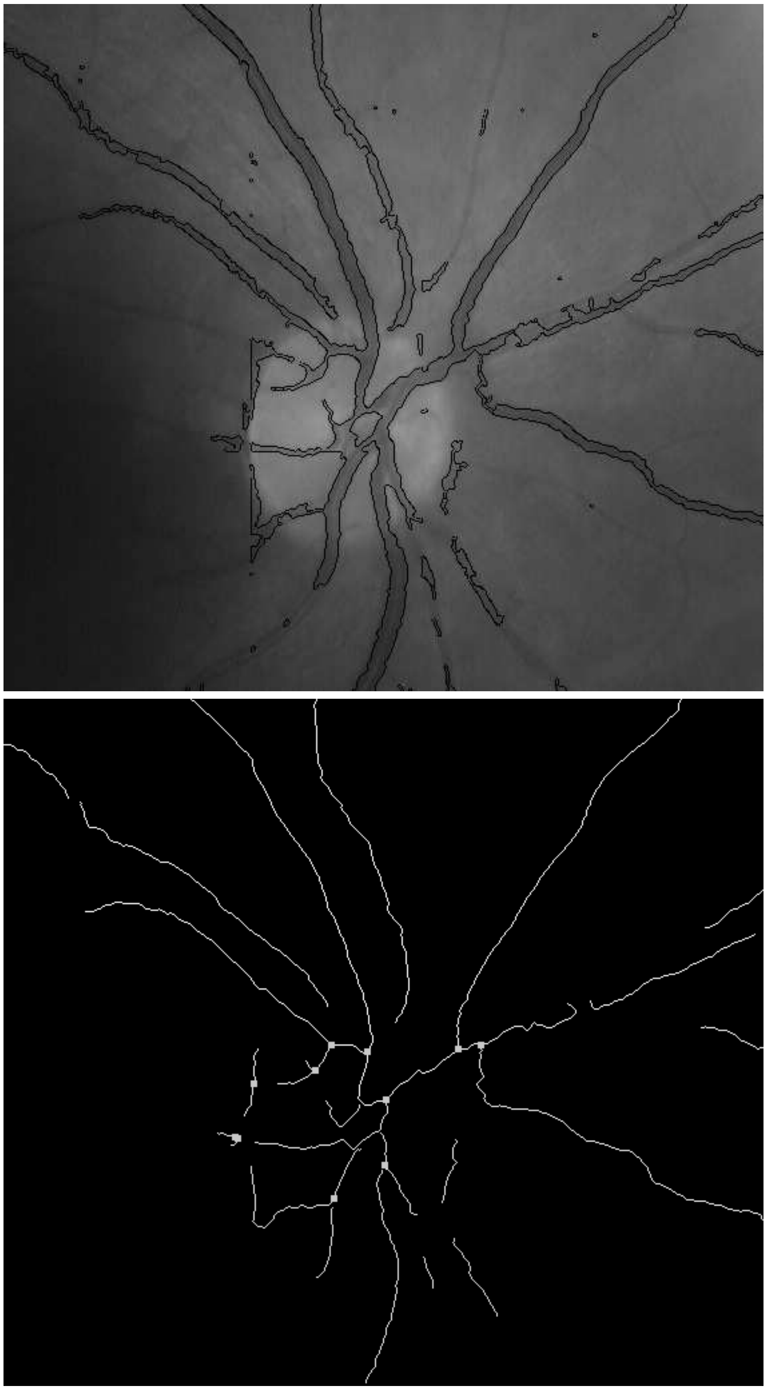

Fig. 5.4: *Left*: Retinal vessel tree (previous to the skeletonisation) over the original image, *Right*: skeleton with feature points used to authenticate

To check the reliability of both authentication systems, a blind test was designed. A set of 100 images were introduced to the system (12 of them belonging to 4 different individuals, acquired in different times). In this test, the matching score for the Cartesian product of the whole set was computed. The diagonal of the Cartesian product contains 1.0 value which corresponds with the comparison of the image with itself.

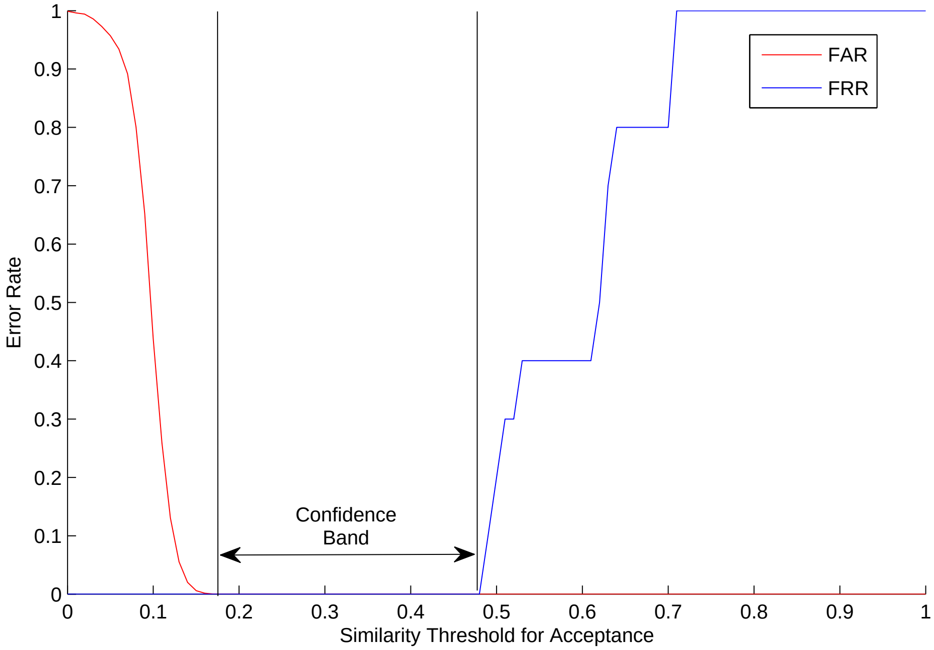

Fig. 5.5: The confidence band of the crease based authentication system [Mariño et al., 2006] can be established around $[0.17, 0.48]$, showing the threshold limits where EER$= 0$ (neither false positives nor false negatives

For the crease based authentication system, the rest of the evaluations proved that values under 0.17 were always obtained from images of different individuals, and over 0.48 from images of the same person (see Fig. 5.5). In the point feature approach, values under 0.3 were always obtained from images of different individuals, and over 0.4 from images of the same person (see Fig. 5.6). The false acceptance rate (FAR) as well as the false rejection (FRR) measure the rate of incorrectly accepted and rejected people, respectively. In both applications, the confidence band shows that the number of false positives and negatives can be reduced to 0 ($FAR = FRR$). In the first application, setting the threshold in values between 0.17 and 0.48, the effectiveness of the system using this database is 100% . In the second application, the confidence band is smaller, but even in that situation, selecting a value between 0.3 and 0.4, the effectiveness of the system is still 100% . Notice that using all the information provided for the whole retinal vessel tree implies a higher confidence band.

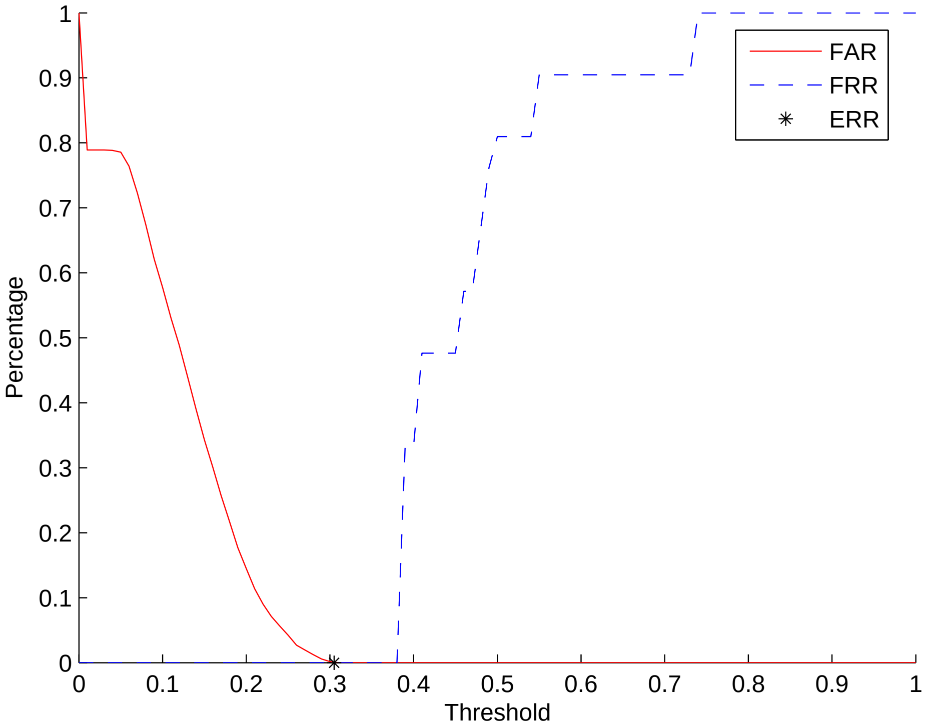

Fig. 5.6: The confidence band of the point feature authentication system [Ortega et al., 2006a] can be established around $[0.3, 0.4]$, showing the threshold limits where EER= 0 (neither false positives nor false negatives)

The mean execution time for the authentication stages (excluding the retinal vessel tree extraction) can be established in about $250ms$ for the point based authentication system. Whereas the extraction of the retinal vessel tree in the original proposal needed several seconds, depending on the image resolution, with the proposed pixel parallel algorithm, the whole authentication process is about 0.44 s. As it has been shown, the retinal vessel tree obtained with the proposed algorithm is suitable to be used in these systems, since the effectiveness is maintained in a 100%.

5.2 Arteriolar-to-Venular Diameter Ratio Estimation

The study and analysis of the vessel geometry features have become the basis of medical applications related with early diagnosis [Brieva et al., 2004] and effective monitoring of therapies in retinopathy [Miles and Nuttall, 1993]. A lot of research has been devoted to the development of strategies for accurate measurement of vessel diameters [Gao et al., 2001a,b], regarding more precise and repeatable diagnostic methods. Among other markers used in early diagnosis, the arteriolar-to-venular (AVR) ratio is used to establish the cardiovascular risk [Hubbard and Brothers, 1999], since it shows the narrowing of the retinal blood vessels. The AVR is determined by measuring the diameters of individual retinal arteriolar and venular calibers. A lower AVR value is associated with a high blood pressure increasing the risk of stroke, diabetes and hypertension. Currently, most of the the medical applications developed to compute the AVR ratio from retinal images are semi-automatic approaches [Caderno et al., 2004, Li et al., 2005], since a manual classification of both types of vessels (vein and artery) should be made in order to compute the AVR ratio. Apart from that, most of the techniques have to deal with the central reflection observed in blood vessels, which corresponds with artery, see Fig. 5.7. This problem is usually corrected by means of a Gaussian model to fit the computed points with the actual ones, like is presented in [Li et al., 2005, Caderno et al., 2004]

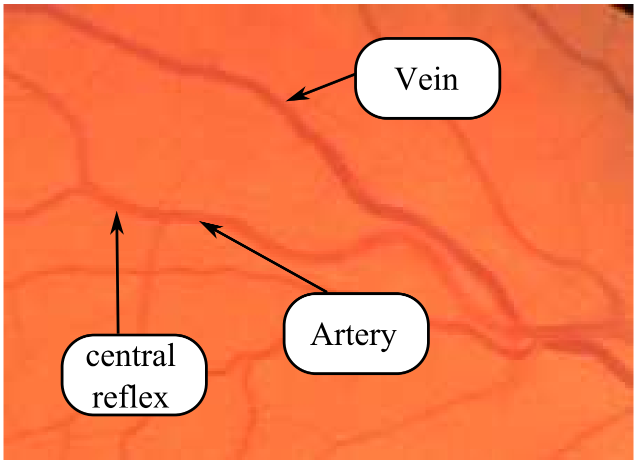

Fig. 5.7: Veins and arteries. Notice the central reflection observed in the artery

The proposed retinal vessel tree extraction algorithm is integrated into a medical application in order to compute the AVR ratio. A specialist classify the vessels (vein or artery) and the AVR ratio is automatically computed from the extracted retinal vessel tree. This work has been presented in [Alonso-Montes et al., 2008b].

5.2.1 AVR ratio computation system

The SIRIUS web application (System for the Integration of Retinal Images Understanding Services) [SIRIUS, 2008], has been developed to be used by specialists from 5 hospitals in Galicia to compute the AVR ratio in angiographies for patient monitoring. Several processes have been defined for the AVR ratio computation, as it is shown in Fig. 5.8.

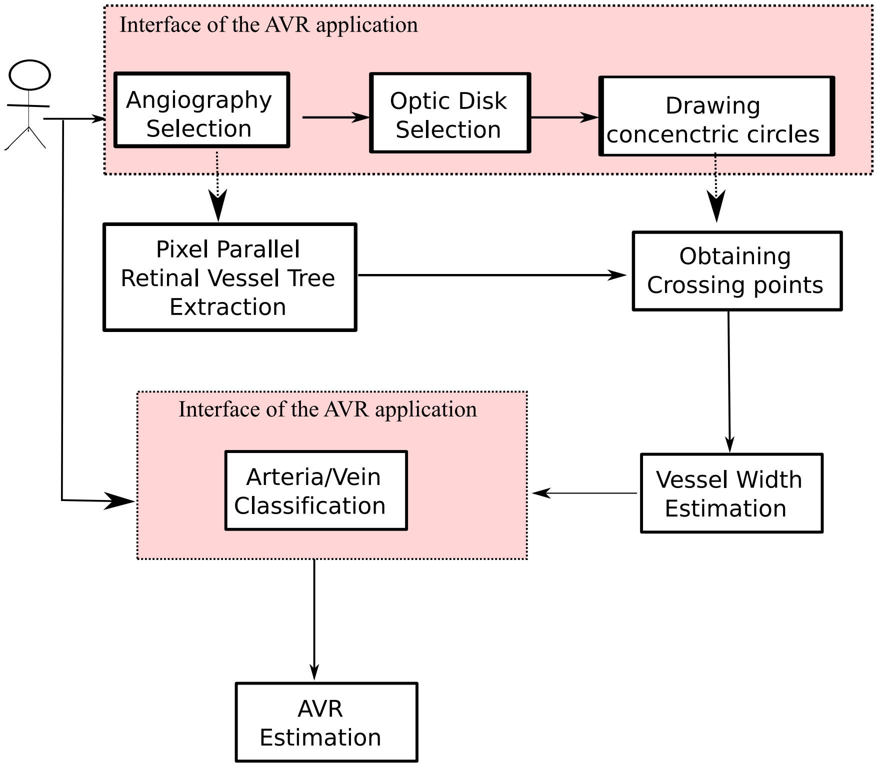
Interface of the AVR application
Angiography Selection
Optic Disk Selection
Drawing concenctric circles
Pixel Parallel Retinal Vessel Tree Extraction
Obtaining Crossing points
Interface of the AVR application
Arteria/Vein Classification
Vessel Width Estimation
AVR Estimation

Fig. 5.8: Flow Diagram of the AVR ratio computation system

Firstly, the specialist selects the retinal image of the patient. Then, this specialist selects the optic disk and three circles, concentric to the optic disk, are drawn. The retinal vessel tree is automatically extracted from the previously selected retinal image. The vessels are measured in the points where the three concentric circles intersect the extracted retinal vessel tree. Finally, the specialist classifies each of those selected points into vein or artery in order to estimate the AVR ratio.

AVR ratio estimation

Once the retinal image of the patient has been selected, the specialist selects the centre of the optic disk over the original image. The system uses this information to draw three concentric circles, d pixels far away from that optic disk (see Fig. 5.9). These concentric circles will mark out a section of a vessel in order to measure its mean width.

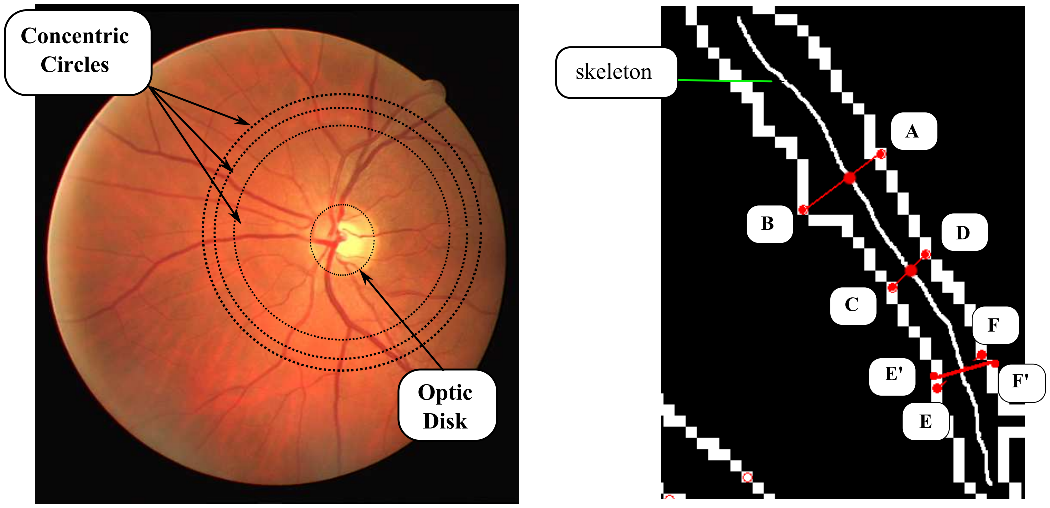

Fig. 5.9: *Left*: Optic disk selected by the specialist, and the three concentric circles. *Right*: Example of the segments that are taken into account to compute the vessel width. *A, B, C, D, E' and F'* are the crossing points between the circles and the extracted retinal vessel tree. Notice that the original *E* and *F* points are discarded, since the segment is not perpendicular to the centre of the vessel (segment joining *E'* and *F'*)

The points, which will be taken into account to estimate the vessel width, are obtained from the intersection of concentric circles, the extracted retinal vessel tree and the centre of the vessel. The segment which join the points must be perpendicular to the centreline of the vessel in order to correctly estimate the vessel diameter. If it is not perpendicular, a perpendicular segment is drawn between the closest points to those ones (see Fig. 5.9).

The estimation of the vessel width is computed as follows:

$$W = \sum_{i=1}^{3} l_i/3 \tag{5.6}$$

where l_i is the length (Euclidean distance) of each of the segments which join the detected intersected points (see Fig. 5.9), and the final width is the mean of these lengths. In Fig 5.9, for example, the l_i distance between points A and B is computed as follows:

$$l_1 = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \tag{5.7}$$

where (x_A, y_A) and (x_B, y_B) are the coordinates of points A and B, respectively.

Once, the vessel diameters are computed, the specialist classify each of those points into venular or arteriolar points. The AVR ratio is computed using the previous classification and according to the following equation:

$$AVR = \frac{\sum W_a}{\sum W_v} \tag{5.8}$$

where W_a and W_v are the widths of arteriolar and venular vessels, respectively, previously computed using Eq. 5.6.

5.2.2 Experimental results

The images used in this experiment have been acquired using a Cannon CR6-45NM Non-Mydriatic Retinal Camera, with a resolution of 768x584 pixels. A total set of 10 images have been used to compute the AVR ratio (2 of them with pathologies). Firstly, the retinal vessel tree is extracted with the proposed algorithm. The main parameters of the algorithm are tuned only once for the whole set of images. The global execution time required to process the whole retinal vessel tree (excluding I/O operations) is 0.1925 s, which is significantly lower compared with the conventional PC-based algorithms. In [Caderno et al., 2004] the active contour is applied only in the area determined by the concentric circles. The whole retinal tree is extracted in [Espona et al., 2007] using the same approach addressed in [Caderno et al., 2004], and it takes about 32.1 s, which is significant higher compared with our technique.

Initially, the optic disk should be established by the specialist in order to draw the concentric circles which will mark out the area to get the points to

No. Image	[Caderno et al., 2004]	Proposed algorithm
1	0.79	0.78
2	0.82	0.84
3	0.81	0.82
4	0.80	0.76
5	0.82	0.80
6	0.89	0.91
7	0.77	0.77
8	0.90	0.93
9	0.83	0.80
10	0.83	0.81

Table 5.1: AVR ratio comparison

estimate the vessel width. These points, which belongs to the intersection of those concentric circles and the extracted retinal vessel tree, will be detected and used to compute the vessel width, according to Eq. 5.6 (see Fig. 5.10, where the selected points are shown with a circle). Finally, a manual classification of venular and arteriolar vessels has been made by an specialist from the CHUS. The AVR ratio is computed following Eq. 5.8, using the widths previously estimated. In this case, those vessel diameters whose length is less than 2 or bigger than 9 have been discarded since they correspond to very small vessels or junctions between vessels, respectively. Other algorithms proposed to estimate the AVR ratio have been also implemented in SIRIUS, particularly the active contour-based proposal addressed in [Caderno et al., 2004]. A comparison between the AVR ratio results computed by means of both proposals are shown in Table 5.1. As it is shown the obtained AVR ratios have an acceptable range of deviation [Couper et al., 2002] (mean average error of 0.002).

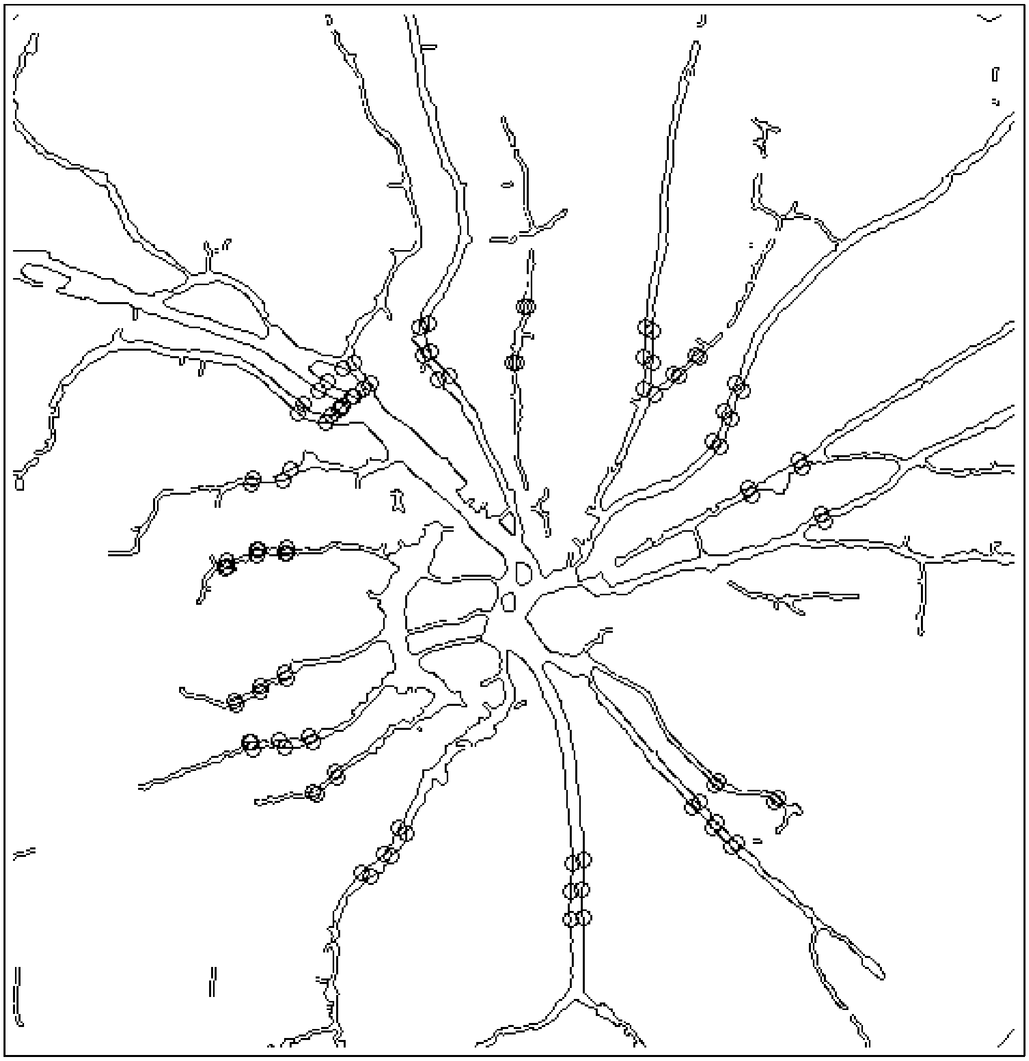

Fig. 5.10: Points selected to compute the AVR ratio

Chapter 6

Discussion and future research

A novel approach for the retinal vessel tree extraction has been presented in this thesis. This algorithm deals with the special features of retinal images at a high computation speed.

From an image processing point of view, the proposed algorithm has been tested with the publicly available Digital Retinal Images for Vessel Extraction database (DRIVE). The maximum average accuracy (MAA) has been computed using the test set provided in the database. The level $MAA = 0.9180$ is good enough for its possible application or integration into different tasks. From the time performance point of view, the algorithm has been successfully implemented in the SCAMP-3 vision system, a chip with SIMD capabilities. The execution time for a standard retinal image of 768x584 pixels from the DRIVE is about 0.1925 s. Although there are proposals in the literature with a better MAA value, the execution time obtained for the proposed retinal vessel tree extraction algorithm is significantly lower (one or several orders of magnitude) than PC-based algorithms. On one hand, the PC industry offers more powerful microprocessors giving to the PC-solutions a more competitive time processing. But, on the other hand, the evolution on processor arrays shows a tendency to increase their size, which balances the distance between PC-based and hardware-based solutions.

Due to the MAA value and the execution time obtained in the previous analysis, the next goal in the thesis was the integration of the algorithm into practical applications. In this case, three applications have been considered, two of them belongs to an authentication system, and another one, related with the medical domain, particularly the AVR ratio estimation system. All those applications have been working using conventional PC-based algorithms for the retinal vessel tree extraction.

In the case of the authentication applications, there two systems to the authentication based on the vessel pattern. Both of them use the skeleton of the retinal vessel tree, so a skeletonisation step has been applied to the final result obtained with the proposed retinal vessel tree algorithm. In the first system, the whole skeleton is used to authenticate the individual. In the second system, only point features (similar to the minutiae) are considered to do the authentication process. The pixel parallel algorithm proposed in this thesis has been integrated in both systems. An analysis of the obtained results showed that the effectiveness of the system is not affected for using the pixel parallel instead of the PC-based approaches.

In the medical domain, usually the accuracy of the results is crucial for the disease diagnosis. In the AVR ratio estimation, the pixel parallel algorithm has been integrated and compared with a contour based approach already implemented in SIRIUS. The results of the measurement of the vessel widths and the AVR computation shows that similar results are obtained.

This algorithm has a good level of accuracy (MAA = 0.9180) and a very low execution time, one or several orders of magnitude less than conventional PC-based applications. In this sense, PC-based systems usually have to get a compromise between accuracy and time performance, which means, more accuracy implies a higher complexity on the operations and more required computation effort.

On the whole, it can be said that the results of the pixel parallel retinal vessel tree extraction algorithm addressed in this thesis can be applied to a wide range of application, tasks and tools, specially those regarding real-time computing (hard or soft real-time requirements) or video-based applications, specially related with tracking. Furthermore, the possibility of projecting of this algorithm onto a focal plane processing systems gives the option to integrate with another devices, such as cameras.

Appendix A

Cellular Neural Networks

Cellular Neural Networks (CNN) [Chua and Yang, 1988] is a computation model which is based on some aspects of neurobiology and adapted to integrated circuits. CNN are a large-scale nonlinear analog circuit which processes signals in real time. It is made of a massive aggregate of regularly spaced circuit clones, called *cells* which communicate with each other directly only through its nearest neighbours. Its continuous time feature allows to manage real-time signal processing in the digital domain and its local interconnection feature make it suitable for VLSI implementations.

A CNN can be outlined as a $2-$,$3-$ or $n-$dimensional array of mainly identical dynamical systems (cells) which satisfies two properties:

- most interactions are local within a finite radius r
- all state variables are continuous valued signals

A.1 CNN cell: the basic unit

The basic unit of CNN is a cell, see Fig. A.1, which contains linear and nonlinear circuit elements. The structure of cellular neural networks is similar to that found in cellular automata: *any cell in a CNN is connected only to its neighbours.* The adjacent cells can interact directly with each other. Cells not directly connected may affect each other indirectly because of the propagation effects of the continuous-time dynamics of the CNN. Theoretically, a CNN of any dimension can be defined. Considering a two dimensional cell, *MxN*

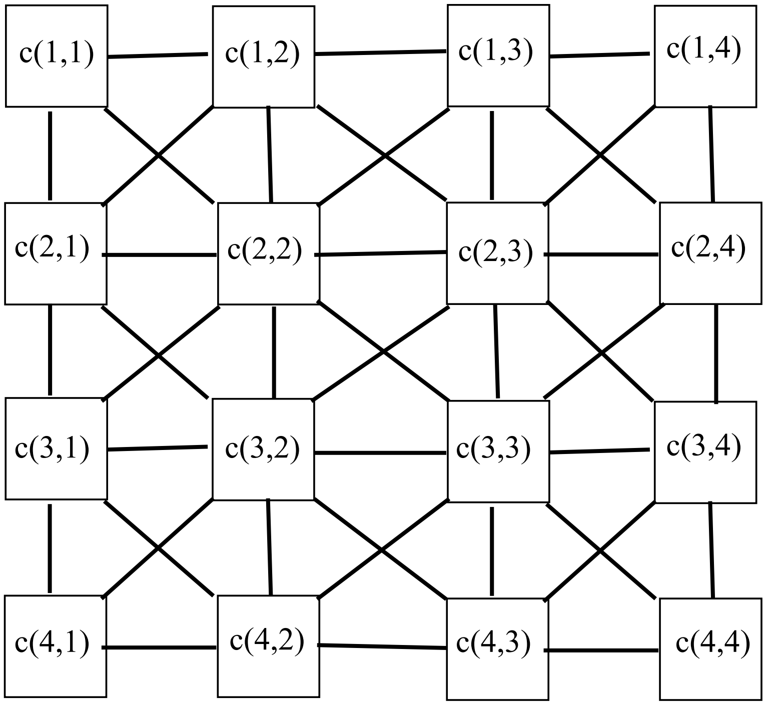

Fig. A.1: Two-dimensional cellular neural network. The circuit size is 4x4. Each circuit unit, called cells, are represented by the squares. The links between the cells indicate the interactions between the cells

CNN, having *MxN* cells arranged in *M* rows and *N* columns, the cell(i,j), *i*-th row and *j*-th column, is denoted by $C(i,j)$. The neighbourhood of $C(i,j)$ (see example in Fig. A.2) is defined by:

$$N_r(i,j) = \{C(k,l)|max\{|k-i|,|l-j| \leq r, l \leq k \leq M; 1 \leq l \leq N\}\} \tag{A.1}$$

where r is a positive integer number. The value $r = 1$ is considered as a 3x3 neighbourhood. This neighbourhood system defined above exhibits a *symmetry* property in the senses that if $C(i,j) \in N_r(k,l)$ then $C(k,l) \in N_r(i,j)$ for all $C(i,j)$ and $C(k,l)$ in a cellular neural network.

A template specifies the interaction between each cell and all its neighbour in terms of their input, state and output variables, see Fig. A.3.

U represents the set of external inputs from the neighbour cells ($U = \{u_{kl}/c(k,l) \in N_r(i,j)\}$). **Y** represents the set of external outputs from the neighbour cells ($Y = \{y_{kl}/c(k,l) \in N_r(i,j)\}$). **B** is the control operator which weights each of the inputs from the neighbour cells of u_{kl} in order to

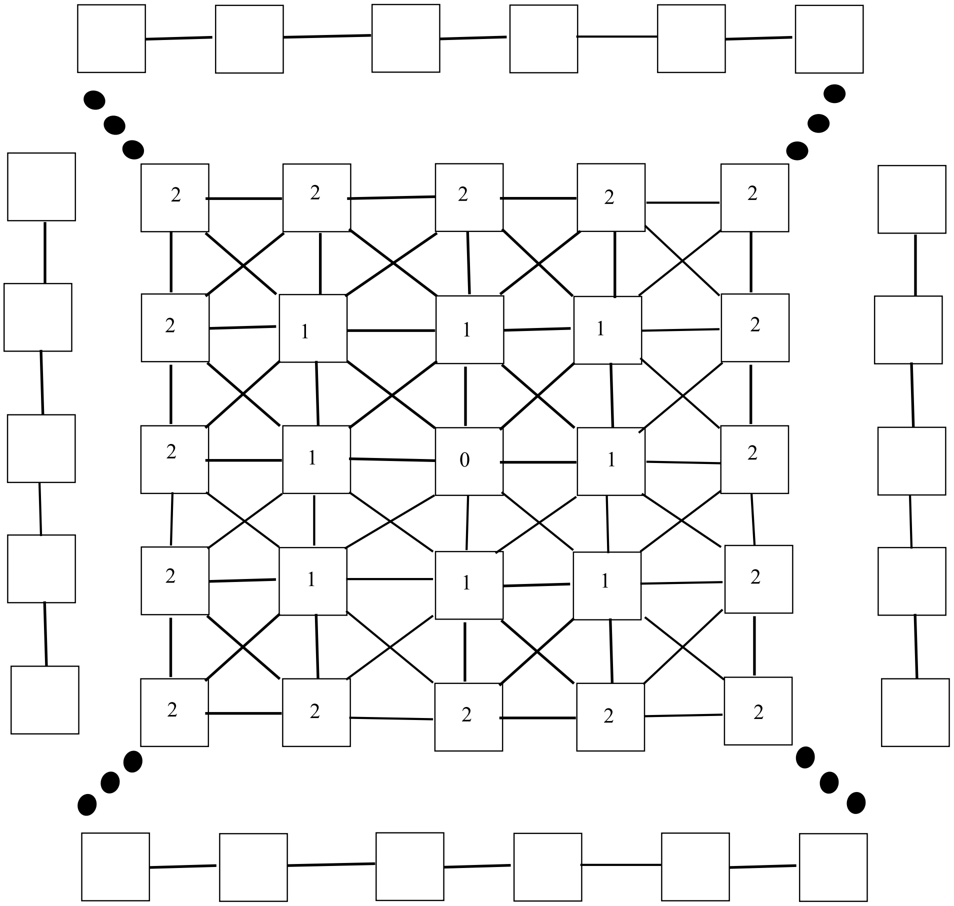

Fig. A.2: Classical scheme of a standard CNN. Each number of the label of each cell is related with the neighbour relationship with respect with the central one (label 0)

weight the importance of their contribution to the new internal state x_{ij} of the cell. **A** is the feedback operator which weights the contributions of the outputs of the neighbour cells y_{kl} to the internal state x_{ij}. **I** is a bias term. It can be a fixed value for each cell or a general for all the network. **G** controls the dynamic of the system. **F** is the activation function which transforms the real value of the internal state x_{ij} into the output y_{ij}.

The dynamic of the model is defined by partial differential equation first order, defining the new state as follows:

$$\frac{d}{dt}x_{ij}(t)0 - x_{ij}(t) + \sum_{k,l\in N_r(i,j)} A_{ijkl}y_{kl}(t) + \sum_{k,l\in N_r(i,j)} B_{ijkl}y_{kl}(t) + I_{ij} \tag{A.2}$$

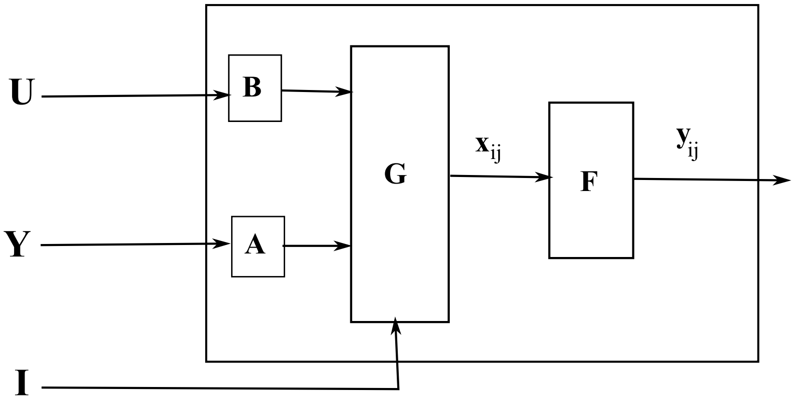

Fig. A.3: Functional scheme of a cell

where ij refers to a grid point associated with a cell on the 2D grid, and $kl \in N_r(i, j)$ is a grid point in the neighbourhood within a radius r of the cell ij.

Since the CNN can calculate practically all types of (planar or spatial convolutions/correlations using a programmable kernel function with a finite spatial window, many existing algorithms and physical phenomena can be translated into analogic CNN algorithms and implemented as complex cells. So, the CNN universal machine was first introduced by [Roska and Chua, 1993]. It is a programmable analog array computer with its own language and operating system. The single chip has the computation power of a supercomputer.

Possible applications of the CNN can be found in [Chua and Yang, 1988], where they were used for noise removal, for feature extraction such as edge or corner detection, even for complex tasks Chinese character recognition. In all those proposals, the CNN works as a two-dimensional filter. Unlike the conventional two-dimensional digital filters, the CNN uses parallel processing of the input image space and delivers its output in continuous time.

Appendix B

CNN-based original approach: experimental results

The CNN-based original algorithm has been tested using 100 retinal images obtained from patients of the central hospital of Santiago de compostela (Complexo Hospitalario Universitario de Santiago, C.H.U.S). The gray level values considered for the computation have been scaled to the range $[-1, 1]$, like in the CNN specifications, where -1 corresponds to white and 1 to black.

All the stages of the proposed algorithm has been applied to each of the sub windows. The parameters used in the processing of all sub windows for each of the stages are shown in Table B.1.

1st Stage	α	β
	0.12	-0.9
3rd Stage	ρ	σ
	0.5	0.5
4th Stage	k_{ext}	k_{int}
	0.4	0.005

Table B.1: Parameters heuristically computed for each of the stages

B.1 Stage 1. Vessel pre-estimation

During this stage, low contrast images are enhanced in order to properly segment vessels. Initially, the histogram equalisation operation improves low contrast images. Then, the adaptive segmentation step, the α and β values are empirically estimated. These values are tuned only once and used for all the images, see values in Table B.1. The optimal threshold value ($thres$) has been computed following the Eq. 3.2. The opening operation removed noise appeared during the adaptive segmentation step. Although noisy points are easily removed, big areas with a similar gray level value, segmented as vessels in the previous step, cannot be removed, compare 3rd and 4th rows in Fig. B.1.

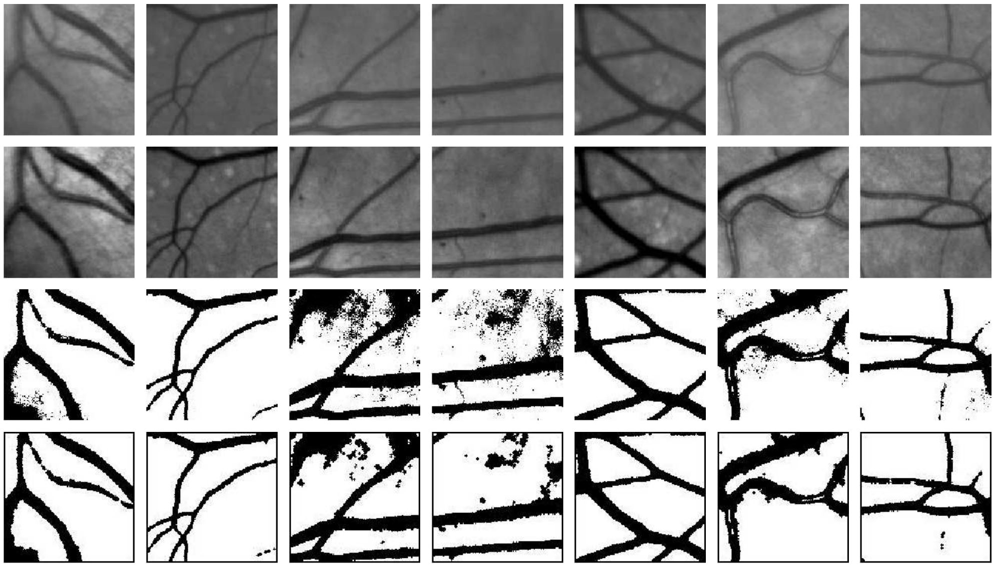

Fig. B.1: **Stage 1**: *1st row* Original image, *2nd row* Histogram equalisation, *3rd row* Adaptive segmentation and *4th row* Opening

B.2 Stage 2. Initial contour estimation

The initial contours have been computed dilating 5 times the regions between the vessels. This number of dilations assure that the contours are completely outside of the vessel locations, removing vessel discontinuities, see Fig. B.2. Then, the binary edge detection step obtains the initial contours needed by PLS.

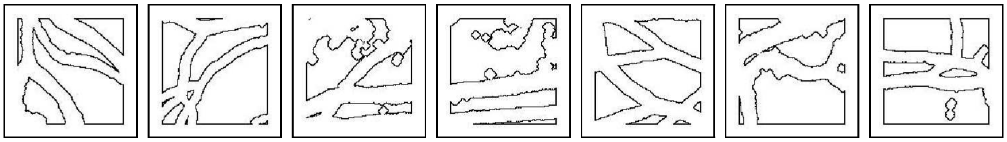

Fig. B.2: Initial contour images

B.3 Stage 3. External potential estimation

The external potential is computed combining the equalised and segmented image obtained during the first stage. The weighting values have been empirically estimated ($\rho = \sigma = 0.5$), see Table. B.1. Notice how the information of vessel continuity is maintained with the combination of both images, see Fig. B.3. The combination of both images smooths the influence of noise and at the same time, in those areas with small discontinuities additional information is added to stop the PLS evolution.

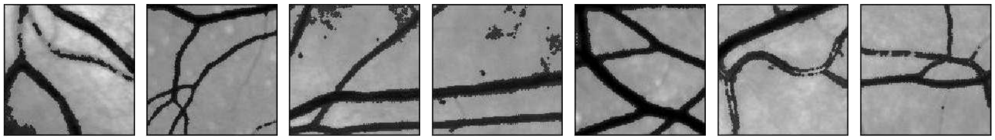

Fig. B.3: External potential images

B.4 Stage 4. PLS evolution

Once the main input images needed by PLS are computed, PLS evolve to fit the exterior of the vessels of the retinal vascular tree. Since the initial contours are situated completely outside of the vessel locations, an inflation balloon potential is needed. The external and internal potential weighting variables ($k_{ext} = 0.4, k_{int} = 0.005$) emphasise the importance of the external potential. Due to the management of the topological changes, merging and splitting regions, noisy regions are removed as well as the vessel topology can be maintained, see Fig. B.4.

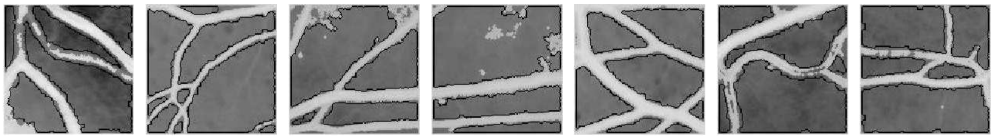

Fig. B.4: PLS superimposed over the inverted external potential image

All the sub windows are joined to get the final retinal vessel tree.

Appendix C

Implementation of the algorithm in the SCAMP

In this chapter the implementation code in the SCAMP is presented. It has been split up into sections corresponding to the main modules of the algorithm. The initialisation of the algorithm includes reading the image from the input device.

C.1 Stage 1. Vessel Pre-estimation

```
//----------------------------------------
// Fast diffusion
//----------------------------------------

NEWS <- A
D <- DIFFUSE

//----------------------------------------
//    Adaptive Threshold
//----------------------------------------
// Substract D from Image Orig

Z <- PIX
K <- D + Z /+ IN(0)
NOP
```

```
Q <- IN(0)
NOP
H <- Q + IN(100)
IF K
  Q <- IN(0)
  NOP
  H <- Q + IN(-100)
ENDIF
// The result is stored in H
```

C.2 Stage 2. Initial Contour Estimation

```
//--------------
//  Erosion
//--------------

Q <- H + IN(-20)
D <-
Z <- Q + IN(0)
NEWS <- Z

Q <- IN(0)
C <-

_load erosion_var, 4
: erosion_loop
// Erosion ==>  -50
IF NEWS
    C <- Q + IN(-50)
IF EAST
    C <- Q +  IN(-50)
IF WEST
    C <- Q + IN(-50)
IF NORTH
    C <- Q + IN(-50)
IF SOUTH
    C <- Q + IN(-50)
ENDIF
```

```
D <- C
NEWS <- D

_sub erosion_var, 1
_jump nz, erosion_loop
// C contains the initial contour
```

C.3 Stage 3. External Potential Estimation

```
//-----------------------------------------
//  SOBEL
//-----------------------------------------

B <- PIX
Q <-
A <- B + Q /image to A

//A = A/2 (with mismatch REDUCTION)
DIV Z+B <- A
Q <- B + A
D <- Q + Z
DIV Z+B <- D
A <- B
//K = A/2 (with mismatch REDUCTION)
DIV Z+B <- A
Q <- B + A
D <- Q + Z
DIV Z+B <- D
K <- B

NEWS <- K
Q<- K
A<-Q+NORTH
B<-Q+SOUTH
Q<-A+B

NEWS<-Q
Z<-WEST
```

```
Q<-Z+EAST
D<-Z+EAST
IF Q
  B <-
  D<-Q+B
ENDIF
NEWS<-K
Q<-K
A<-Q+EAST
B<-Q+WEST
Q<-A+B
NEWS<-Q
Z<-NORTH
Q<-Z+SOUTH
A<-Z+SOUTH
IF A
  B <-
  A<-Q+B
SF
B<-D+A;
NEWS <- B

//==============================
// Range change of Sobel result
//==============================

Q <-
A <- B + Q
D <- B + Q

_load sumVar, 5
: change_range
K <-  A + D
A <- K
_sub sumVar, 1
_jump nz change_range
// Result stored in A

//------------------------------------------------
// 3/4*Sobel + 1/4*Iseg
//------------------------------------------------
```

```
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
D <- Z


DIV K+Z <- D
Q <- Z + D
Z <- Q + K
DIV K+Q <- Z
K <- Q
A <- D + K

// Result stored in A
//-------------------------------------------------

DIV K+Z <- H
D <- Z + H
Q <- D + K
DIV K+Z <- Q
H <- Z

DIV K+Z <- H
D <- Z + H
Q <- D + K
DIV K+Z <- Q
K <- Z

D <- A
A <- D + K
B <- NEWS // Sobel value
Z <- A + B
A <- Z
//------------------------------------------
// diffusion of A
//------------------------------------------
_load ldiff2, 2 //diffusion time
```

```
: diff_sob_loop2

// diffuse EAST
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + EAST
A <- Z

// diffuse SOUTH
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + SOUTH
A <- Z

// diffuse WEST
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + WEST
A <- Z

// diffuse NORTH
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
```

```
K <- Z

Z <- K + NORTH
A <- Z

_sub ldiff2, 1
_jump nz, diff_sob_loop2

//end of  diffusion
//-----------------------------------------
// Result stored in Z

Z <- A + B
A <- Z
Z <- A + B
A <- Z
Z <- A + B
A <- Z
```

C.4 Stage 4. PLS evolution

```
_load cpd, 1
_load sobel,0
_load tdiff,1


: main
_load cloop, 8 // evolutions per frame
_load ball, 68 //balloon offset (zero)
_load balloon,100 // balloon strength

_load pls_cond, 2
/=====================================
/ initialisation of images
/Load and Threshold Initial CONTOUR Image
/B = EXTERNAL POTENTIAL
B <- Z
```

```
/C = CONTOUR (REGION)

D <- C
H <- D
K <- D
Q <-
C <- H + K + Q

/======================================
: pls_start

//contour evolution loop
: main_loop

H <- IN (0)

// find contour from region
NOP
K <- H + IN (-90)
//K = 3x3 dilation of not C
NEWS <- C; SF
IF EAST
  K <- H + IN(90)
SF
IF WEST
  K <- H + IN(90)
SF
IF NORTH
  K <- H + IN(90)
SF
IF SOUTH
  K <- H + IN(90)
SF
IF NEWS
  K <- H + IN(-90)
ENDIF


/invert region
A <- H + IN (30)
IF C
```

```
  A <- H + IN (-30)
ENDIF
C <- A


/------------------------------------------------------------------
/    IPE module
//Q - Result of Contour Diffusion

A <- H + IN (-50)
IF K
  A <- H + IN (50)
ENDIF


// diffusion of A
_load ldiff, tdiff
: diff_loop

// diffuse EAST
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + EAST
A <- Z

// diffuse SOUTH
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + SOUTH
A <- Z
```

```
// diffuse WEST
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + WEST
A <- Z

// diffuse NORTH
DIV K+Z <- A
D <- Z + A
Q <- D + K
DIV K+Z <- Q
NEWS <- Z
K <- Z

Z <- K + NORTH
A <- Z

_sub ldiff, 1
_jump nz, diff_loop

Q <- Z + H + IN (-50)


/-------------------------------------------------------------------
/ BPE module - calculate balloon potential to A


_load test, balloon
_sub balloon, ball
_sub test, balloon
_sub test, balloon
_load balloon, test

_output, balloon, IN
NOP
D <- H + IN
```

```
_output, ball, IN
NOP

IF C
  D <- H + IN
ENDIF

A <- D

/------------------------------------------------------------------
/Total Potential to D:

D <- B + A + Q

: skip

/------------------------------------------------------------------
// Contour Evolution NORTH

Q <- C

/can expand
K <- IN (-100)

/only if neighbour
NEWS <- C
IF SOUTH
  K <- IN (100)
ENDIF

/only if decr. potential
NEWS <- D
A <- D + SOUTH
NEWS <-
Z <- A + NEWS
IF Z
  K <- IN (100)
ENDIF

_add cpd,1
```

```
_sub cpd,1
_jump nz skip1
/only if no collision
Z <- Q
NEWS <- Z
A <- IN (-100) /danger
IF Z
  A <- IN (100) /NO danger
SF

IF NORTH
  A <- IN (100) /NO danger
//if danger around - cancel move
SF

IF A
  K <- IN (100)
ENDIF
Z <- A
NEWS <- Z
IF EAST
  K <- IN (100)
SF

IF WEST
  K <- IN (100)
ENDIF
SF

: skip1

/expand
IF K
  Q <- IN (35)
ENDIF

/-------------------------------------------------------------------
// Contour Evolution SOUTH

/can expand
K <- IN (-100)
```

```
/only if neighbour
NEWS <- C
IF NORTH
  K <- IN (100)
ENDIF

/only if decr. potential
NEWS <- D
A <- D + NORTH
NEWS <-
Z <- A + NEWS
IF Z
  K <- IN (100)
ENDIF

_add cpd,1
_sub cpd,1
_jump nz skip2
/only if no collision
Z <- Q
NEWS <- Z
A <- IN (-100) /danger
IF Z
  A <- IN (100) /NO danger
SF

IF SOUTH
  A <- IN (100) /NO danger
//if danger around - cancel move
SF

IF A
  K <- IN (100)
ENDIF

Z <- A
NEWS <- Z
IF EAST
  K <- IN (100)
SF
```

```
IF WEST
  K <- IN (100)
SF

ENDIF
: skip2
/expand
IF K
  Q <- IN (35)
ENDIF

/-------------------------------------------------------------------
// Contour Evolution EAST

/can expand
K <- IN (-100)

/only if neighbour
NEWS <- C
IF WEST
  K <- IN (100)
ENDIF

/only if decr. potential
NEWS <- D
A <- D + WEST
NEWS <-
Z <- A + NEWS
IF Z
  K <- IN (100)
ENDIF

_add cpd,1
_sub cpd,1
_jump nz skip3
/only if no collision
Z <- Q
NEWS <- Z
A <- IN (-100) /danger
IF Z
```

```
  A <- IN (100) /NO danger

SF

IF EAST
  A <- IN (100) /NO danger
//if danger around - cancel move
SF

IF A
  K <- IN (100)
ENDIF
Z <- A
NEWS <- Z
IF NORTH
  K <- IN (100)
SF

IF SOUTH
  K <- IN (100)
SF

ENDIF
: skip3

/expand
IF K
  Q <- IN (35)
ENDIF

/------------------------------------------------------------------
// Contour Evolution WEST

/can expand
K <- IN (-100)

/only if neighbour
NEWS <- C
IF EAST
  K <- IN (100)
ENDIF
```

```
/only if decr. potential
NEWS <- D
A <- D + EAST
NEWS <-
Z <- A + NEWS
IF Z
  K <- IN (100)
ENDIF

_add cpd,1
_sub cpd,1
_jump nz skip4
/only if no collision
Z <- Q
NEWS <- Z
A <- IN (-100) /danger
IF Z
  A <- IN (100) /NO danger
SF

IF WEST
  A <- IN (100) /NO danger
//if danger around - cancel move
SF

IF A
  K <- IN (100)
ENDIF

Z <- A
NEWS <- Z
IF NORTH
  K <- IN (100)
SF

IF SOUTH
  K <- IN (100)
ENDIF
: skip4
```

```
/expand
IF K
  Q <- IN (35)
ENDIF

/-------------------------------------------------------------------
/ update contour

C <- H + IN (-30)
IF Q
  C <- H + IN (30)
ENDIF

_sub cloop, 1
_jump nz main_loop


// Done. Final region is in C

NOP
K <- H + IN (-90)
//K = 3x3 dilation of not C
NEWS <- C; SF
IF EAST
  K <- H + IN(90)
SF
IF WEST
  K <- H + IN(90)
SF
IF NORTH
  K <- H + IN(90)
SF
IF SOUTH
  K <- H + IN(90)
SF
_sub pls_cond, 1
_jump nz hole_fill

_jump finished

//-----------------------------------------------------------------
```

```
//  STAGE 5. Hole Filling
//------------------------------------------------------------------

: hole_fill
/A = Image
Z <- K + IN (0)

_input slider, #80

_load slider, 40

A <- IN (50)
NOP
_output slider, IN
NOP
Q <- Z + IN
IF Q
  A <- IN (-50)
ENDIF

Z <- IN (-50)
NOP
D <- IN(-100)
NOP
NEWS <- IN (100)
IF Z + EAST
  D <- IN (100)
ENDIF
NOP
IF Z + WEST
  D <- IN (100)
ENDIF
NOP
IF Z + NORTH
  D <- IN (100)
ENDIF
NOP
IF Z + SOUTH
  D <- IN (100)
ENDIF
NOP
```

```
//Reconstruction
_load a, 120
: fill_loop

//3x3 dilation
SF
NEWS <- D; SF
IF EAST
  D <- IN(100)
SF
IF WEST
  D <- IN(100)
SF
NEWS <- D; SF
IF NORTH
  D <- IN(100)
SF
IF SOUTH
  D <- IN(100)
SF

//C and A
D <- IN (-100)
ENDIF

_sub a, 1
_jump nz, fill_loop
Q <-
Z <- D + Q
C <- Z
_load cpd, 0
_load sobel,0
_load tdiff,1



_load cloop, 40
_load ball, 68
_load balloon, 72
@
```

```
_load pls_cond, 1
_jump pls_start

//------------------------------------------------------------------
//   Contour edge detection
//------------------------------------------------------------------
: finished
//  Final region is in C
// to display - find contour from region
NOP
K <- H + IN (-90)
//K = 3x3 dilation of not C
NEWS <- C; SF
IF EAST
  K <- H + IN(90)
SF
IF WEST
  K <- H + IN(90)
SF
IF NORTH
  K <- H + IN(90)
SF
IF SOUTH
  K <- H + IN(90)
SF
IF NEWS
  K <- H + IN(-90)
ENDIF

OUT K
```

Bibliography

M. Al-Rawi and H. Karajeh. Genetic algorithm matched filter optimization for automated detection of blood vessels from digital retinal images. *Computer Methods and Programs in Biomedicine*, 87(3):248–253, 2007.

M. Al-Rawi, M. Qutaishat, and M. Arrar. An improved matched filter for blood vessel detection of digital retinal images. *Comput. Biol. Med.*, 37 (2):262–267, 2007. ISSN 0010-4825.

C. Alonso-Montes, D. L. Vilarino, and M. G. Penedo. CNN-based Automatic Retinal Vascular Tree Extraction. In *IEEE Int. Workshop on Cellular Neural Networks and their Applications (CNNA)*, pages 61–64, 2005a.

C. Alonso-Montes, D. L. Vilarino, and M. G. Penedo. On the Automatic 2D Retinal Vessel Extraction. In *LNCS Pattern Recognition and Image Analysis)*, volume 3681, pages 165–173, 2005b.

C. Alonso-Montes, P. Dudek, D. L. Vilarino, and M. G. Penedo. On Chip Implementation of a Pixel Parallel Approach for Retinal Vessel Tree Extraction. In *European Conf. on Circuit Theoriy and Design (ECCTD 2007)*, pages 511–514, 2007.

C. Alonso-Montes, M. Ortega, M. G. Penedo, and D. L. Vilariño. Pixel Parallel Vessel Tree Extraction for a Personal Authentication System. In *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pages 1569 –1599, 2008a.

C. Alonso-Montes, M. G. Penedo, and D. L. Vilariño. Arteriolar-to-Venular Diameter Ratio Estimation: A Pixel-Parallel Approach. *Int. Workshop on Cellular Neural Networks and their Applications (CNNA)*, 2008b.

C. Alonso-Montes, D. L. Vilariño, P. Dudek, and M. G. Penedo. Fast Retinal Vessel Tree Extraction: A Pixel Parallel Approach. *Int. Journal of Circuit Theory and Application*, 2008c.

Anafocus. Anafocus co., seville, spain. `http://www.anafocus.com/`, 2008.

D. R. W. Barr, S. J. Carey, A. Lopich, and P. Dudek. A Control System for a Cellular Processor Array. In *IEEE Int. Workshop Cellular Neural Networks and their Applications*, pages 176–181, 2006.

A. Blake and M. Isard. Active Contours. *Robotica, Springer-Verlag London*, 17:459–462, 1999.

Ruud Bolle, Sharath Pankanti, and A. K. Jain. *Biometrics, Personal Identification in Networked Society: Personal Identification in Networked Society.* Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0792383451.

V.M. Brea, M. Laiho, D.L. Vilariño, A. Paasio, and D. Cabello. A Binary-based On-chip CNN Solution for Pixel-Level Snakes. *Int. Journal of Circuit Theory and Applications*, 2006.

J. Brieva, M. Galvez, and C. Toumoulin. Coronary extraction and stenosis quantification in X-ray angiographic imaging. In *IEEE-EMBC Proc. Int. Conf. Eng. Med. Biology Society*, volume 1, pages 1714–1717, 2004.

I. G. Caderno, M. G. Penedo, C. Mariño, M. J. Carreira, F. Gomez-Ulla, and F. González. Automatic Extraction of the Retina AV Index. In *LNCS Int. Conf. Image Analysis and Recog. (ICIAR'04)*, volume 3212, pages 132–140, 2004.

A. Can, C. V. Stewart, B. Roysam, and H. L. Tanenbaum. A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(3):347–364, 2002.

V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22:61–79, 1997.

T. Chanwimaluang and G. Fan. An efficient blood vessel detection algorithm for retinal images using local entropy thresholding. In *Proc. Int. Symp. Circuits Syst. (ISCAS '03)*, volume 5, pages 21–24, May 2003.

S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of Blood Vessels in Retinal Images using Two-Dimensional Matched Filters. *IEEE Trans. Med. Imag.*, 8:263–269, 1989.

L. O. Chua and T. Roska. The CNN paradigm. *IEEE Trans. Circuits Syst.*, 40:147–156, 1993.

L. O. Chua and Lin Yang. Cellular Neural Networks: Theory. *IEEE Trans. Circuits Syst.*, 35(10):1257–1272, Oct 1988.

CLS. Cnn software library. version 1.1. analogic and neural comput. lab. computer and automation research inst. of the hungarian academy of sciences. `http://lab.analogic.sztaki.hu/Candy/csl.html`, 2007.

L. D. Cohen and I. Cohen. Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:1131–1147, 1993.

D. J. Couper, R. Klein, L. D. Hubbard, and T. Y. Wong. Reliability of retinal photography in the assessment of retinal microvascular characteristics: the atherosclerosis risk in the communities study. *American Journal of Ophtalmology*, 133:78–88, 2002.

P. A. Van den Elsen, J. B. A. Maintz, E. J. D. Pol, and M. A. Viergever. Automatic registration of CT and MR brain images using correlation of geometrical features. *IEEE Trans. Med. Imag.*, 14(2):384–396, 1995.

Drive. Digital retinal images for vessel extraction. `http://www.isi.uu.nl/Research/Databases/DRIVE/`, 2008.

P. Dudek. Adaptive sensing and image processing with a general-purpose pixel-parallel sensor/processor array integrated circuit. In *Int. Workshop Computer Architectures for Machine Perception and Sensing (CAMPS)*, pages 18–23, September 2006.

P. Dudek and D. L. Vilariño. A Cellular Active Contours Algorithm Based on Region Evolution. In *IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pages 269–274, 2006a.

P. Dudek and D. L. Vilariño. A Cellular Active Contours Algorithm Based on Region Evolution. In *IEEE Int. Workshop on Cellular Neural Networks and their Applications*, pages 269–274, 2006b.

L. Espona, M. J. Carreira, M. Ortega, and M. G. Penedo. A Snake for Retinal Vessel Segmentation. In *LNCS Pattern Recognition and Image Analysis*, volume 4478, pages 178–185, 2007.

H. Eviatar and R. L. Somorjai. A fast, simple active contour algorithm for biomedical images. *Pattern Recogn. Lett.*, 17(9):969–974, 1996.

M. Foracchia, E. Grisan, and A. Ruggeri. Detection of optic disc in retinal images by means of a geometrical model of vessel structure. *IEEE Trans. Med. Imag.*, 23(10):1189–1195, 2004.

X.W. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. Thom. A Method of Vessel Tracking for Vessel Diameter Measurement on Retinal Images. In *IEEE Int. Conf. Image Processing*, volume 2, pages 881–884, 2001a.

X.W. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. Thom. Measurement of Vessel Diameters on Retinal for Cardiovascular Studies. In *Medical Image Understanding and Analysis*, 2001b.

M. Goldbaum, S. Moezzi, A. Taylor, S. Chatterjee, J. Boyd, E. Hunter, and R. Jain. Automated diagnosis and image understanding with object extraction, object classification, and inferencing in retinal images. *Proc. Int. Conf. Image Processing*, 3:695–698, 1996.

R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzdky. Fast geodesic active contours. *IEEE Trans. Image Processing*, 10:1467–1475, 2001.

F. Gomez-Ulla, M. I. Fernandez, F. Gonzalez, P. Rey, M. Rodriguez, M. J. Rodriguez-Cid, F. F. Casanueva, M. A. Tome, J. Garcia-Tobio, and F. Gude. Digital Retinal Images and Teleophthalmology for Detecting and Grading Diabetic Retinopathy . *Diabetes Care*, 25(8):1384–1389, 2002.

C. Heneghan, J. Flynn, M. O'keefe, and M. Cahill. Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis. *Medical Image Analysis*, 6(23):407–429, 2002.

D. Hillier, V. Binzberger, D. L. Vilariñño, and Cs. Rekeczky. Topographic Cellular Active Contour Techniques: Theory, Implementations and Comparisons. *Int. Journal of Circuit Theory and Appl.*, 34(2):183–216, 2006.

A. Hoover and M. Goldbaum. Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. *IEEE Trans. Med. Imag.*, 22(8):951–958, 2003.

A. Hoover, V. Kounznetsova, and M. Glodbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med. Imag.*, 19(3):203–210, 2000.

L. D. Hubbard and R. J. Brothers. Methods for evaluation of retinal microvascular abnormalities associated with hypertension/sclerosis in the atherosclerosis risk in communities studies. *Survey of Ophtamology*, 106: 2269–2280, 1999.

Luo Hui, Lu Qiang, R.S. Acharya, and R. Gaborski. Robust snake model. In *IEEE Proc. Int. Conf. Comput. Vision and Pattern Recog.*, volume 1, pages 452–457, 2004.

L. Huiqi and O. Chutatape. Automatic location of optic disk in retinal images. *Int. Conf. Image Processing*, 2:837–840, 2001.

H. F. Jelinek, M. J. Cree, J. J. G. Leandro, J. V. B. Soares, R. M. Cesar-Jr., and A. Luckie. Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy. *Journal of the Optical Society of America*, 24(5):1448–1456, 2007.

Xiaoyi Jiang and D. Mojon. Adaptive Local Thresholding by Verification-Based Multithreshold Probing with Application to Vessel Detection in Retinal Images. *IEEE Trans. Pattern Anal. Machine Intell.*, 25:131–137, 2003.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contours Models. *Int. Journal on Computer Vision*, 1:321–331, 1988.

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

R. A. Kirsch. Computer Determination of the Constituent Structure of Biological Images. *Comp. Biomed. Res.*, 4(3):315–328, 1971.

I. Kompatsiaris, D. Tzovaras, V. Koutkias, and M. G. Strintzis. Deformable boundary detection of stents in angiographic images. *IEEE Trans. Med. Imag.*, 19(6):652–662, 2000.

D. Koozekanani, K. Boyer, and C. Roberts. Retinal thickness measurements from optical coherence tomography using a Markov boundary model. *IEEE Trans. Med. Imag.*, 20(9):900–916, 2001.

D. Koozekanani, K. L. Boyer, and C. Roberts. Tracking the optic nervehead in OCT video using dual eigenspaces and an adaptive vascular distribution model. *IEEE Trans. Med. Imag.*, 22(12):1519–1536, 2003.

L. Lam, S. W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(9):869–885, 1992.

H. Li, W. Hsu, M. L. Lee, and T. Y. Wong. Automatic Grading of Retinal Vessel Caliber. *IEEE Trans. Bio-Med. Eng.*, 52:1352 – 1355, 2005.

Qin Li, D. Zhang, Lei Zhang, and P. Bhattacharya. A New Approach to Automated Retinal Vessel Segmentation Using Multiscale Analysis. *Int. Conf. Pattern Recognition*, 4:77–80, 2006.

R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(2):158–175, 1995.

C. Mariño, M. G. Penedo, M. Penas, M. J. Carreira, and F. Gonzalez. Personal authentication using digital retinal images. *Pattern Anal. and Applications*, 9:21–33, 2006.

M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, A. A. Bharath, and K. H. Parker. Retinal Blood Vessel Segmentation by Means of Scale-Space Analysis and Region Growing. In *Medical Image Computing and Computer-Assisted Intervention*, pages 90–97, 1999.

M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, N. Chapman, A. A. Bharath, and K. H. Parker. Retinal Vascular Tree Morphology: A Semi-Automatic Quantification. *IEEE Trans. Biomed. Eng.*, 49: 912–917, 2002.

G. K. Matsopoulos, P. A. Asvestas, N. A. Mouravliansky, and K. K. Delibasis. Multimodal registration of retinal images using self organizing maps. *IEEE Trans. Med. Imag.*, 23(12):1557–1563, 2004.

T. McInerney and D. Terzopoulos. Topologically Adaptable Snakes. In *Proc. 5th IEEE Int. Conf. Computer Vision*, pages 840–845, 1995.

T. McInerney and D. Terzopoulos. Deformable Models in Medical Image Analysis. In *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 171–180, 1996a.

T. McInerney and D. Terzopoulos. Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation. *IEEE Trans. Med. Imag.*, 18: 840 – 850, 1999.

T. McInerney and D. Terzopoulos. Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*, 1:91–108, 1996b.

F. P. Miles and A. L. Nuttall. Matched filter estimation of serial blood vessel diameters from video images. *IEEE Trans. Med. Imag.*, 12:147–152, 1993.

A. Mosquera, R. Dosil, V. Leboran, F. Pardo, F. Vonez-Ulla, and B. Hayik. ART-VENA: Retinal vascular calibre measurement. In *Lecture Notes in Computer Science, IbPRIA 2003*, volume 2652, pages 598–605, 2003.

C. Heneghan N. Ryan and P. de Chazal. Registration of digital retinal images using landmark correspondence by expectation maximization. *Image and Vision Computing*, 22:883–898, 2004.

R. Nekovei and Ying Sun. Back-propagation network and its configuration for blood vessel detection in angiograms. *IEEE Trans. Neural Networks*, 6(1):64–72, 1995.

M. Niemeijer, J.J. Staal, B. van Ginneken, M. Loog, and M.D. Abramoff. Comparative study of retinal vessel segmentation methods on a new publicly available database. In J. Michael Fitzpatrick and M. Sonka, editors, *SPIE Medical Imaging*, volume 5370, pages 648–656. SPIE, SPIE, 2004.

J. O'Brien and N. Ezquerra. Automated Segmentation of Coronary Vessels in Angiographic Image Sequences Utilizing Temporal, Spatial Strucctural Constraints. In *Proc. Conf. Visualization in Biomedical Computing. SPIE*, 1994.

M. Ortega, C. Mariño, M. G. Penedo, M. Blanco, and F. Gonzalez. Biometric Authentication using digital retinal images. In *Int. Conf. WESEAS on Applied Computer Science*, pages 422–427, 2006a.

M. Ortega, C. Mariño, M. G. Penedo, M. Blanco, and F. Gonzalez. Personal Authentication Based on Feature Extraction and Optica Nerve Location in Digital Retinal Images. In *WSEAS Tans. on Computers*, volume 5, pages 1169–1176, 2006b.

S. Osher and J. A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *J. of Comp. Physics*, 79:12–49, 1988.

S. Ourselin and Li Rongxin. Extension of Deformable Models: Hybrid Approaches for Analysis of Medical Images. In *IEEE-EMBS Proc. Int. Conf. Eng. Med. Biol. Soc.*, pages 7182–7185, 2005.

N. Patton, T. M. Aslam, T. MacGillivray, I. J. Deary, B. Dhillon, R. H. Eikelboom, K. Yogesan, and I. J. Constable. Retinal image analysis: concepts, applications and potential. *American Journal of Ophthalmology*, 141(3): 99 – 127, 2006.

L. Pedersen, M. Grunkin, B. Ersboll, K. Madsen, M. Larsen, N. Christoffersen, and U. Skands. Quantitative measurement of changes in retinal vessel diameter in ocular fundus images. *Pattern Recognition Letters*, 21 (13-14):1215–1223, 2000.

R. Perfetti, E. Ricci, D. Casali, and G. Costantini. Cellular Neural Networks With Virtual Template Expansion for Retinal Vessel Segmentation. *IEEE Trans. Circuits Syst. II*, 54:141–145, 2007.

A. Pose-Reino, F. Gomez-Ulla, B. Hayik, M. rodriguez Fernandez, M. J. Carreira-Nouche, A. Mosquera-Gonzalez, M. G. Penedo, and F. Gude. Computerized measurement of retinal blood vessel calibre:description, validation and use to determine the inluence of ageing and hypertension. *Journal of Hypertension*, 23, 2005.

C. Rekeczky and L. O. Chua. Computing with Front Propagation: Active Contour and Skeleton Models in Continuous-Time CNN. *J. VLSI Signal Process. Syst.*, 23(2-3):373–402, 1999.

C. Rekeczky, A. Schultz, I. Szatmari, T. Roska, and L. O. Chua. Image Segmentation and Edge Detection via Constrained Diffusion and Adaptive Morphology: a CNN approach to Bubble/debris Image Enhancement. In *Proc. 6th Int. Symp. Nonlinear Theory and its Applications (NOLTA '97)*, pages 209–212, Dec 1997.

T. Roska and L. O. Chua. The CNN Universal Machine: An Analogic Array Comuter. *IEEE Trans. Circuits Syst. II*, 40(3):163–173, March 1993.

H. Schmitt, M. Grass, V. Rasche, O. Schramm, S. Haehnel, and K. Sartor. An X-ray-based method for the determination of the contrast agent propagation in 3-D vessel structures. *IEEE Trans. Med. Imag.*, 21(3):251–262, 2002.

SIRIUS. SIRIUS: System for the Integration of Retinal Images Understanding Services. `http://varpa.lfcia.org/RetinalResearch.htm`, 2008.

J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree. Retinal Vessel Segmentation using the 2-D Gabor Wavelet and Supervised Classification. *IEEE Trans. Med. Imag.*, 25:1214–1222, 2006.

M. Sofka and C. V. Stewart. Retinal Vessel Centerline Extraction Using Multiscale Matched Filters, Confidence and Edge Measures. *IEEE Trans. Med. Imag.*, 25(12):1531–1546, 2006.

J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge-Based Vessel Segmentation in Color images of the Retina. *IEEE Trans. Med. Imag.*, 23:501–509, april 2004.

STARE. STructured Analysis of the REtina. `http://www.ces.clemson.edu/~ahoover/stare/`, 2008.

M. Tico and P. Kuosmanen. Fingerprint matching using an orientation-based minutia descriptor. *IEEE Trans. Pattern Anal. Machine Intell.*, 25 (8):1009–1014, 2003.

R. Toledo, X. Orriols, P. Radeva, X. Binefa, J. Vitria, C. Canero, and J.J. Villanueva. Eigensnakes for vessel segmentation in angiography. In *Proc. Int. Conf. Pattern Recognition*, volume 4, pages 340–343, 2000.

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

E. Trucco and P. J. Kamat. Locating the optic disk in retinal images via plausible detection and constraint satisfaction. *Int. Conf. Image Processing*, 1: 155–158, 2004.

U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. In *Proc. IEEE*, volume 92, pages 948– 960, June 2004.

F.L. Valverde, N. Guil, J. Munoz, Q. Li, M. Aoyama, and K. Doi. A deformable model for image segmentation in noisy medical images. In *Proc. Int. IEEE Conf. Image Processing*, volume 3, pages 82–85, 2001.

D. L. Vilariño and P. Dudek. Evolution of Pixel Level Snakes towards an efficient hardware implementation. In *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pages 2678–2681, 2007.

D. L. Vilariño and C. Rekeczky. Shortest Path Problem with Pixel Level Snakes: Application to Robot Path Planning. In *IEEE Int. Conf. Cellular Neural Networks and their Applications*, 2004a.

D. L. Vilariño and C. Rekeczky. Pixel-Level Snakes on the CNNUM: Algorithm Design, On-Chip Implementation and Applications. *Int. Journal of Circuit Theory and Applications*, 33:17–51, 2005.

D. L. Vilariño and C. Rekeczky. Implementation of a Pixel-Level Snake Algorithm on a CNNUM-based Chip Set Architecture. *IEEE Trans. Circuits Syst. I*, 51:885–891, 2004b.

D. L. Vilariño, D. Cabello, X. M. Pardo, and V. M. Brea. Pixel-Level Snakes. In *Proc. Int. Conf. Pattern Recognition*, volume 1, pages 640–643, 2000.

D. L. Vilariño, D. Cabello, X .M. Pardo, and V. M. Brea. Cellular neural networks and active contours: a tool for image segmentation. *Image and Vision Computing*, 21:189–204, Feb 2004.

T. Walter, J. C. Klein, P. Massin, and A. Erginay. A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina. *IEEE Trans. Med. Imag.*, 21 (10):1236–1243, 2002.

T. Y. Wong, R. Klein, and J. M. Tielsch. Retinal microvascular abnormalities, and their relation to hypertension, cardiovascular diseases and mortality. *Survey of Ophtamology*, 46:59–80, 2001.

T. Y. Wong, R. Klein, A. R. Sharrett, B. B. Duncan, D. J. Couper, J. M. Tielsch, B. E. K. Klein, and L. D. Hubbard. Retinal arteriolar narrowing and risk of coronary heart disease in men and women: The Atherosclerosis Risk in Communities Study. *American Journal of Ophthalmology*, 134(2): 1153–1159, 2002.

J. Xu, E. Sung, O. Chutatape, Ce Zheng, and P. Chew. Automated optic disk segmentation via a modified snake technique. *Int. Conf. Control, Automation, Robotics and Vision*, pages 1–6, 2006.

F. Zana and J. C. Klein. A multimodal registration algorithm of eye fundus images using vessels detection and Hough transform. *IEEE Trans. Med. Imag.*, 18(5):419–428, 1999.

L. Zhou, M. S. Rzeszotarski, L. J. Singerman, and J. M. Chokreff. The Detection and Quantification of Retinopathy using Digital Angiograms. *IEEE Trans. Med. Imag.*, 13(4):619–626, 1994.