Fully automatized parallel segmentation of the optic disc in retinal fundus images

Daniel Díaz-Pernil[a], Irene Fondón[b], Francisco Peña-Cantillana[c], Miguel A. Gutiérrez-Naranjo[c]

[a] CATAM Research Group - Department of Applied Mathematics I, University of Seville, Spain
[b] Department of Signal Theory and Communications, University of Seville, Spain
[c] Research Group on Natural Computing - Department of Computer Science and AI, University of Seville, Spain

Keywords:

Optic disc
Hough transform Parallel
image processing GPU

A B S T R A C T

This paper presents a fully automatic parallel software for the localization of the optic disc (OD) in retinal fundus color images. A new method has been implemented with the Graphics Processing Units (GPU) technology. Image edges are extracted using a new operator, called *AGP-color segmentator*. The resulting image is binarized with Hamadani's technique and, finally, a new algorithm called *Hough circle cloud* is applied for the detection of the OD. The reliability of the tool has been tested with 129 images from the public databases DRIVE and DIARETDB1 obtaining an average accuracy of 99.6% and a mean consumed time per image of 7.6 and 16.3 s respectively. A comparison with several state-of-the-art algorithms shows that our algorithm represents a significant improvement in terms of accuracy and efficiency.

1. Introduction

Image analysis and processing have great significance in the field of medicine, especially in non-invasive treatment and clinical study. However, with the development of new technologies, larger quantity of data, especially high quality images, is available. Therefore, there is a new necessity of efficient and fast algorithms capable of processing and extracting meaningful features from images in a reasonable time. This is the case of mass screening programs for the early detection of retinal diseases such as glaucoma or diabetic retinopathy. Visual inspection of the large number of images so obtained is a time consuming task for the medical experts. Moreover, CAD (Computer Aided Diagnosis) tools based on retinal image processing developed in the past are limited by the balance between accuracy and complexity due to their sequential programming.

The OD is seen on fundus color photographs as a bright yellowish disc in human retina from where the blood vessels and optic nerves emerge. Its relevance resides in the fact that it is a key point for the diagnosis of a wide variety of diseases such as glaucoma or diabetic retinopathy. Moreover, it is usually taken as a base for detecting other anatomical structures (macula, blood vessels) and retinal abnormalities (microaneurysms, hard exudates, drusens, etc.). Most of the methods found in the literature are semi-automatized. This means that the computer treatment is crucial in the localization and detection of the OD, but the human expert is who takes the final decision. In this paper, a fully automatized method is presented where no human expert is necessary for the detection of the OD.

Changes in the OD can indicate the current state and progression of a certain disease while its diameter is usually used as a reference for measuring retinal distances and sizes [1]. Therefore, accurate OD localization and detection of its boundary is a principal and basic step for automated diagnosis systems [2].

In the literature, the number of methods that take into account the vectorial nature of the color retinal image is limited. Instead, its gray level representation or a single color plane are adopted. For instance, in [3] and [4], the gray level image is thresholded in order to retain the brightest pixels of the image. From them, the location where all vessels converge is encountered and denoted as the OD center. In [5], the grey level version of the color original image is used to segment the OD on two of the three different approaches presented: multi-thresholding and active contour without edges. In [6], the authors also use images in gray scale. Instead of segmenting the OD, several points delimiting them are provided. The system needs to be previously trained with a set of labelled images.

Intensity (I) plane from the HIS color space is selected in [7]. The method applies an adaptive local contrast enhancement technique to retinal images and identifies the position of OD by variance measurement between the intensity of each pixel and intensities of adjacent pixels.

Morphology techniques are adopted in [8] with a preprocessing step added at the beginning to correct illumination and enhance contrast. Next, the OD region is found through a morphological operation. The relative constancy of the position between the fovea and the OD is utilized in [9], locating both regions at once. Other image processing techniques frequently used in the literature are gradient vector flow [10] and graph cuts [11], always applied on a single color plane.

A well-known set of methods based on the processing of the red (R) color plane is ARGALI (Automatic cup-to-disc Ratio measurement system for Glaucoma detection and AnaLysIs) [12–14]. These methods share a common structure that comprise the location of a region of interest (ROI) by the automatic selection of 0.5% of the brightest pixels of the image, OD segmentation with a variational level set technique and contour refinement with an ellipse fitting final stage. Trying to improve the ARGALI system, AGLAIA (Automatic Glaucoma Diagnosis and Its Genetic Association Study through Medical Image Informatics) system introduces a cascade of modified level set steps [15]. Another method using the R color plane is presented in [16]. The authors segment the cup and the OD by using Candy method and circular Hough to recognize the initial disk.

Green (G) plane is the image color channel selected by the authors in [9,17–19]. More precisely, in [18], the geometric relationship between the OD and main blood vessels is utilized to identify the disc location. The described method is based on a fuzzy voting mechanism. In [17], a method based on Hoover's algorithm is presented. In this approach, the location of the OD is approximated by searching for regions of high intensity, diversity of gradient directions, and convergence of blood vessels. The method of [19] is based on a pyramidal decomposition of the G plane image and the use of the Hausdorff distance based template matching. In [20], the OD is not segmented, it is only detected. In this approach, the generalized Hough transform and the G or R plane are used.

A less frequent color space is used in [21] where the gray level version of color retinal image is processed along with the u-coordinate from the UCS color coordinate system. The method localizes the OD by detecting the vessels in the image and backtracking them to their origin.

Only a few methods make use of the three planes of a certain color space, usually RGB along with HSI family, for its simplicity on the first case and its intensity separation on the second one. The methods proposed in [22] and [23] are based on the independently processing of the R, G and B (blue) planes of RGB color space. In [22], a system based on PCA analysis is proposed. In [23], color segmentation is carried out in the original RGB image with a statistical classification method used to detect the objects with bright yellowish color. A group of features are defined as non-intersecting classes in the feature space and then they can be easily identified with a classifier. In [24], the authors presented the idea of detecting the OD by an entropy filter applied to the image expressed in the HSI color space.

The a priori knowledge about OD shape has propitiated the development of algorithms that search for circles in the image with the well-known Hough transform. This is a feature extraction technique that tries to find imperfect instances of objects within a certain class of shapes by a voting procedure. G color plane form RGB color space is the most widely used image representation in these kind of methods [25–30]. In this way, [25] used the repeated thresholding technique on G to find the brightest areas in the image. Afterwards, they introduced the roundness of the object to detect OD features and finally localized the OD by using the Hough transform. However, gray level image is used by other techniques such as in [26] or [31].

In [32], the OD is detected by using the three color planes separately. The method used is KNN and it is trained by a set of 500 images and considers vessel analysis. We would like to highlight the method presented in [33], where a color space is used for stereo color images. Such method uses a KNN classifier and the training images are preprocessed in parallel.

The above mentioned Hough based methods forget the color content of the image and need some parameters to be experimentally fixed such as the minimum and maximum radius values. The elimination of these fixed parameters would be translated into a prohibitive increase in computational time. To overcome these two drawbacks while improving the quality of the results, we propose an algorithm that considers the complete RGB color space for image processing and does not need any parameter to be fixed. This second point is achieved due to its implementation in parallel instead of sequential that makes the technique very fast and leads it to deal with high complexity with a low consumed time. In this paper, the proposed algorithm detects shapes which correspond to inexact or approximate circles in an image. Such technique is based on the classical Hough Transform (HT), but instead of choosing a circle candidate, it takes a set of circumferences (a cloud). The set of pixels on such circumferences are a region in the image. The skeleton of such image is the obtained OD. Such skeleton is not a perfect circle and fit the shape of the OD better than an exact circle. The computation of the circle cloud requires a computational time which is not admissible for sequential computers. This drawback is sorted in this paper by using GPU. This architecture allows to implement a parallel software which with competitive average accuracy and computational time.

Graphics Processing Units (GPU) have emerged as general-purpose coprocessors in recent years. Traditionally designed for gaming applications, GPUs offers many computing threads arranged in a Single-program Multiple-data (SPMD) model. The chosen hardware architecture for our parallel implementation has been the Compute Unified Device Architecture, (CUDA^{TM}), which allows the parallel NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a sequential Central Processing Unit (CPU).

The choice of this parallel architecture is supported by several reasons. Firstly, CUDA^{TM} allows programmers a friendly model for implementing easily parallel programs. From the practical side, there exists an increasing interest in the specialized industry for the development of more and more powerful GPUs which can be used for general purposes. This interest leads, on the one hand, to a more economically accessible (and hence, more extended) hardware and, on the other hand, to the development of more powerful computational units. The use of this new parallel architecture is currently explored as a tool for paralleling the treatment of digital images [34,35].

Finally, we would like to emphasize the OD obtained in this paper is not a perfect circle, as it is usual in the most of papers from above. We try to adapt our results to the real shape of the OD that appear in the different images used in our experiment. So, we go a step further to obtain the best segmentation approach.

The paper is organized as follows: Firstly, we recall some basic aspects of the Hough transform. Next we give a brief description of our algorithm and some experimental results on the GPU implementation. The paper finishes with some final remarks and conclusions.

2. Hough circle clouds

The *Hough transform* is a well-known feature extraction technique used in image analysis. The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses [36–38]. The voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

The basic idea behind the Hough transform is to convert the image into a parameter space that is constructed specifically to describe the desired shape analytically. Maxima in this parameter space correspond to the presence of the desired shape in image space. As OD can be considered as an imperfect circle on a fundus image, the usual mathematical parametrization as $(x-a)^2+(y-b)^2=r^2$ (where (a, b) are the coordinates of the center of the circle that passes through (x, y) and r is its radius) is used. Each circle can be identified as a point (a, b, r) in a 3D parametric space, and each point (a, b, r) receive votes which are accumulated in an accumulator array for all parameter combinations. The accumulation space presents a high value when an imperfect circle is found in the original image.

The search of circles with the Hough transform is based on a precise radius value, that is, only circles with the desired radius will be detected. State-of-the-art techniques usually introduce this value as an experimentally fixed parameter [28,31]. Obviously, the use of experimental parameters could lead to erroneous results and must be avoided.

In order to avoid these drawbacks, in this paper we propose a new technique called Hough Circle Cloud for the detection of the OD. It exploits the massive parallelism of the GPU technology. First of all, the algorithm searches for the best radius value within a wide range of possible ones. Searching the proper radius in a range of feasible candidates is performed in a competitive time due to the massive parallelization of the voting procedure of the Hough transform implemented on a GPU. Since the OD is not a perfect circle, many different circles are proposed as approaches to the OD by the use of the Hough transform. The set of pixels of all these circumferences is usually a thick irregular ring including the real limits of the OD. This irregular ring is called the Hough Circle Cloud. All the pixels in the Hough circle cloud correspond to an approach to the OD. The proposed method provides an skeletonization of this Hough Circle Cloud via an adaptation of the Hall and Guo method. This skeletonization is an imperfect circle obtained from the set of circles proposed as candidates by the Hough transform which captures the irregularities of the real OD.

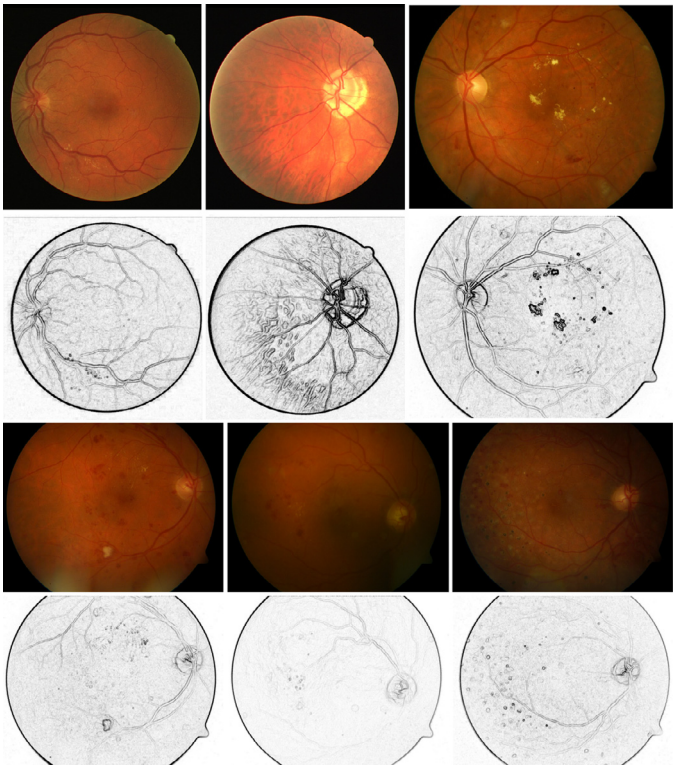


Fig. 1. Original images (up, in color) and segmented ones using the proposed AGP edge detector (down, in grey scale).

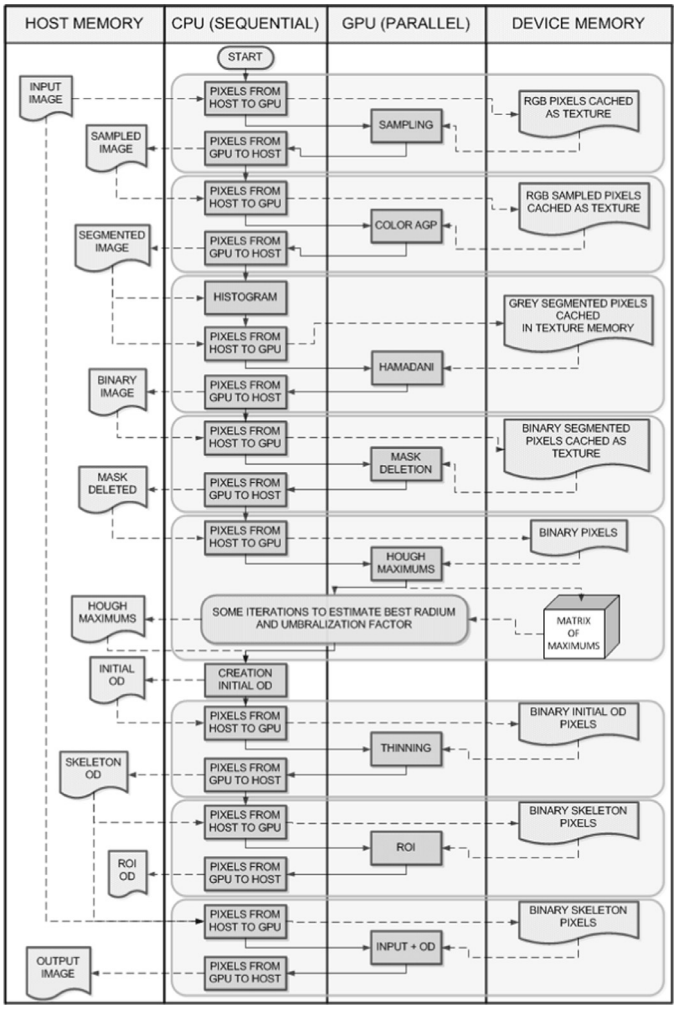


Fig. 2. Workflow of the algorithm working on a GPU card.

3. Algorithm of the detection of the optic disc

Next, a description of the different phases of the algorithm is provided. A detailed workflow can be seen in Fig. 2. The algorithm takes the image as an input and it outputs the location and border of the OD. The key feature of the process is that it is fully automatic. No human expert is needed in any step. The structure of our algorithm to detect the OD is the following:

- **Phase 1: Compression of the input image.** It could be an unexpected step when you are working in parallel with GPUs, but the reason is found in the phase 2. We work for each pixel of

the input image with a neighborhood of pixels whose size is fixed. The compression is performed by sampling. In this way, the new image is not distorted with respect to the original one. Firstly, the image is split into squares of four connected pixels and secondly, the upper left pixel is taken from each square. Each image of size $a \times b$ with $\max\{a, b\} > 1200$ is compressed, but the new maximum is never less than 600.

In our case study, the images of DIARETDB1 (whose size is 1500 $\times$ 1152 pixels) have been compressed, not the ones from the database DRIVE (whose size is 565 $\times$ 584 pixels).

- **Phase 2: Segmentation (Color AGP Segmentator).** In order to detect image edges, a variant of the usual gradient operator has been implemented. Our software can consider neighborhoods of 3×3, 5×5, $\ldots(2n+1) \times (2n+1)$ (where $(2n+1) \times (2n+1)$ is the size of the square of pixels centered in the chose pixel). In the current study, a 5×5 neighborhood is used in order to control the blur effect. A new improved algorithm has been developed for this paper from the previous ideas presented in [39] that we called Color AGP Segmentator. This new operator is based on the changes of the intensity of the color of the image. Each plane from the RGB original color image is independently processed (regardless which of them is the most informative). The three color planes from the RGB color space are considered and the final result for each pixel corresponds to the highest value obtained from the planes R, G and B. In this way, the possible saturation problems are sorted (Fig. 1).

  The basic idea is to work with more information data than the 3×3 Sobel operator, but to be more efficient than 5×5 Sobel operator, always from a parallel processing point of view. We consider to work with a dynamical perspective. Firstly, for each pixel, we take the possible four directions and look for the appropriate direction. Later, we control the efficiency of our algorithm considering to work with only 12 pixels (the 5×5 Sobel operator uses 20 pixels).

  So, given a digital image with n^2 pixels ($n \in \mathbb{N}$), we define a tissue P system with cellular division whose input is given by the set $\{a_{ij} : 1 \leq i, j \leq n\}$ where $a \in RGB$.

  Next, we give outline of how we can obtain a new approximation of the intensity gradient function *AGP operator*, of an image using tissue P systems with cell division.

  The bio-inspired algorithm (tissue P systems) consists on the following stages:

  1. Generating stage: The system creates the necessary number of copies of the pixels for the following stage. To do this, the P system uses *the environment* and *division rules*.
  2. Choosing a direction stage: The system decides among four directions: west-east, northwest-southeast, north-south and northeast-southwest.
  3. Gradient stage: Chosen a direction, the system approximates the intensity gradient function.
  4. Output stage: The system sends to the output cell the results of the previous stage.

  The bio-inspired algorithm is described in the Appendix (a detailed description can be found at [39]).
- **Phase 3: Binarization (Hamadani algorithm).** In order to choose an appropriate threshold to the binarization, a recent implementation [40] of the Hamadani algorithm [41] is applied. In this way, a linear combination of the mean and the standard deviation of the values of pixels of the images is used. Fig. 3 shows the grey scale images of Fig. 1 after the binarization.
- **Phase 4: Mask deletion.** In order to avoid erroneous results, the obtained image is processed by eliminating the eye border. This is performed by applying a threshold on each color plane of the original image with the algorithm presented in [40]. All the pixels with zero value on the three color channels are considered as background in our final binarized image. Examples of these resulting black and white images from Fig. 3 are shown in Fig. 4 (Top).
- **Phase 5: Computing of Hough maximums.** Initially, the black connected components in the binarized image are checked as candidates for the OD. To this aim, firstly the circular Hough transform is applied in parallel to the image in an interval of radius wide enough to consider all the possible OD. The interval is automatically chosen by the software according to the image size. The distance between two consecutive circle is one pixel. This phase is crucial in the drastic time improvement of this method with respect to others found in the literature.

  The Hough transform is based in a voting process which is performed for each black pixel of the image. Our algorithm processes the black pixels of the image in a sequential way, but the voting processing is parallel on the set of pixels.

  For each radius in the interval, the Hough transform is performed in parallel via a computation matrix. At this stage, a threshold is sequentially chosen. This threshold will filter the values of the matrix Hough which a reasonable number of points on a circumference.

  One of the main novelties of this method is that the concept of *cloud*. Instead of searching a maximum value, our software

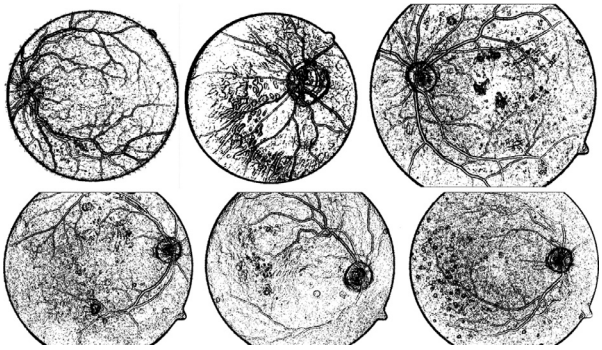


Fig. 3. Binarized images from Fig. 1.

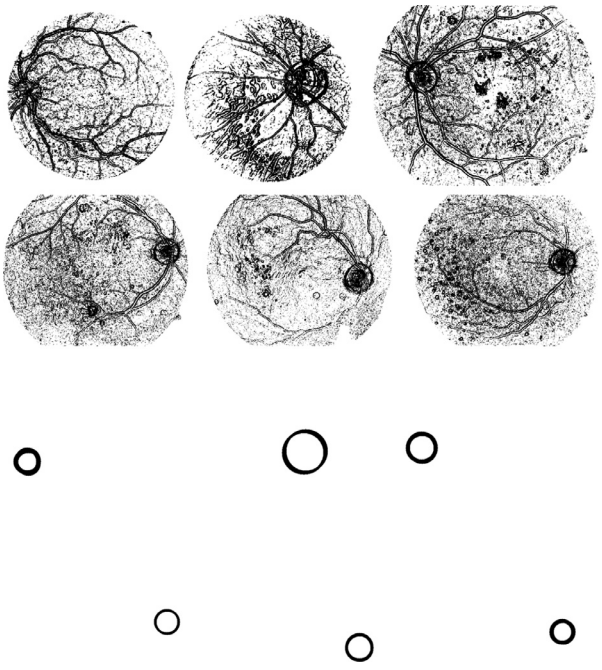


Fig. 4. (Top) Images from Fig. 3 without eye border. (Bottom) Initial ODs of such images.

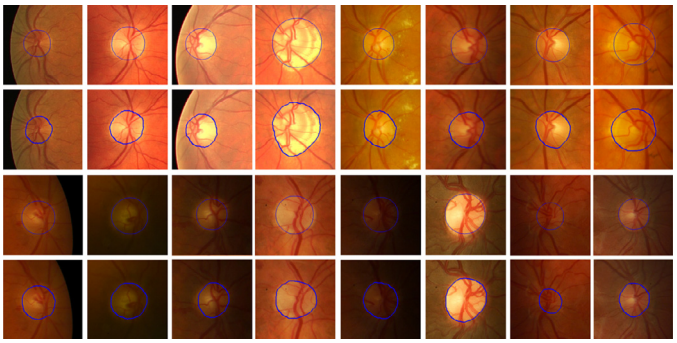


Fig. 5. Several Examples. Our results vs. the ground truth provided by experts.

consider *all* the radius which surpass a percentage of successes. Such percentage is iteratively searched in order to consider a representative cloud of values.
The graphical representation of the cloud is a set of circumferences instead of a single one (see Fig. 4 bottom). The software also considers the case where the set of circumferences are disjoint. In this case, only one set of circumferences is chosen after a process of erosion and dilation.

- **Phase 6: Creation of OD.** The initial OD is a circle set by the maximum on the Hough voting procedure. Some examples of candidates to OD resulting from the images appear in Fig. 4. These candidates are thinned with an improved version of the bio-inspired implementation of the skeletonizing Hall & Guo algorithm presented in [42], where cellular automata are used for the implementation. Since the searched boundary of the OD is one-pixel width, the frames are closed considering 4-adjacency. Fig. 5 shows some examples of the obtained OD. Rows 1 and 3 show the result obtained with our algorithm. Rows 2 and 4 show the ground truth provided by experts. The OD border is marked in blue on the original image.

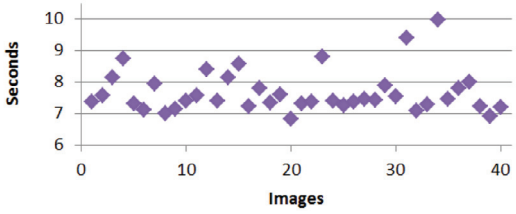
4. Experimental results

Our algorithm has been tested on two different publicly available databases i.e DRIVE [43] and DIARETDB1 [44]. The used metrics are the usual in the literature. *Performance* is the percentage of images where the OD is found. The other metrics are the following (based on true and false positives and true and false negatives):

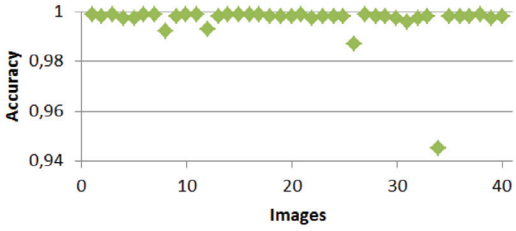
$$Oratio = \frac{tp}{tp+fp+fn} \qquad Sensitivity = \frac{tp}{tp+fn}$$

$$Accuracy = \frac{tn+tp}{tp+fp+tn+fn} \qquad Specificity = \frac{tn}{tn+fp}$$

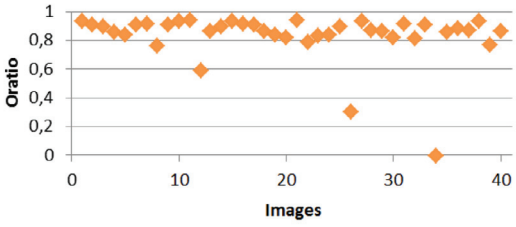
In order to compare it with other algorithms found in the literature, several statistical data (Execution times, Accuracy, Oratio, Sensitivity and Specificity) have been computed for all images of the two databases. The results are shown in Figs. 6 and 7. To sum up, Table 1 shows the average values for the statistics given for each database. Let us remark that the fully automatized detection of the OD in the DRIVE database (40 images) is performed in 5.1 min with an average accuracy of 99.6 %. For the DIARETDB1 database, 89 images were fully processed in 24.2 min and 99.6% average accuracy. Finally, Table 2 shows a comparison of our algorithm with other methods found in the literature. Let us remark that such table does not include the computational time needed for processing the images since it is not usually provided by the authors.



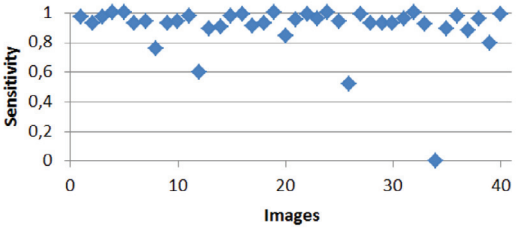
(a) Execution time



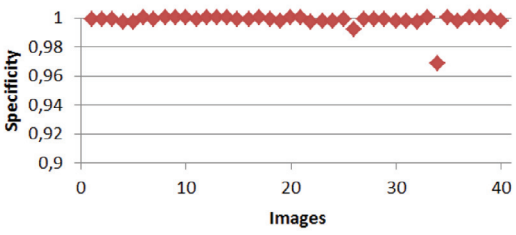
(b) Accuracy



(c) Oratio

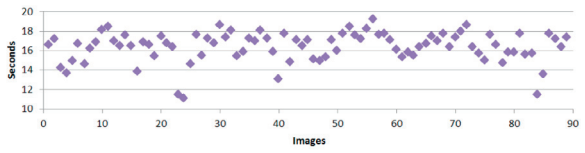


(d) Sensitivity

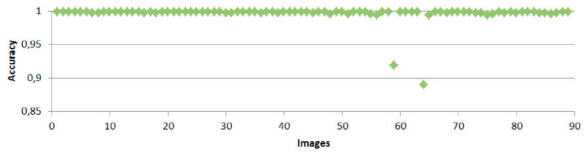


(e) Specificity

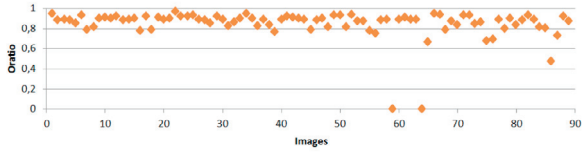
Fig. 6. Statistics for DRIVE database.



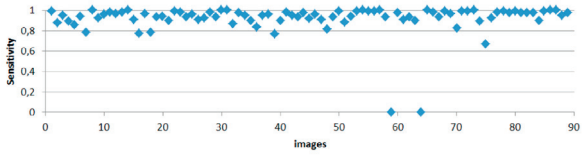
(a) Execution time



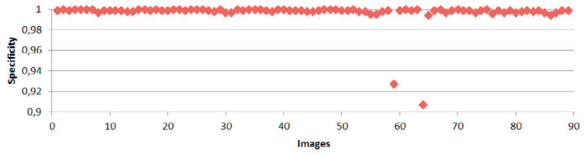
(b) Accuracy



(c) Oratio



(d) Sensitivity



(e) Specificity

Fig. 7. Statistics for DIARETDB1.

4.1. Graphics Processing Units

GPUs are high performance computing many-core processors which has become available for compute intensive applications in the last years. In particular, NVIDIA has developed the CUDA^{TM} technology which allows to program efficiently GPUs for general purposes. CUDA^{TM} technology is based on a Single Instruction, Multiple Threads (SIMT) programming model i.e., the same instruction is executed simultaneously on many data elements by different threads. They are especially well-suited to address problems that can be expressed as data-parallel computations since GPUs devote more transistors to data processing than data caching and flow control.

General programming recommendations are optimizing load balancing and increasing processor occupancy. However, depending on the algorithm structure, both recommendations cannot be applied simultaneously. Thus, some kind of tradeoff must be undertaken, since an optimally balanced implementation may increase the use of registers and the need for sharing data among threads, which decreases occupancy. Moreover, parallelization becomes even more challenging, if the algorithm presents workload-dependent computations and non-linear memory references. The former produces divergence among threads, if the layout is not carefully planned. The latter affects the locality of references, which entails serialized memory accesses.

Table 1
Statistics for databases DRIVE and DIARETDB1.

	DRIVE	DIARETDB1
Size (number of images)	40	89
Number of failed images	1	2
Performance (%)	97,5	97,7
Average Oratio (%)	83,3	84,3
Average Sensitivity (%)	89,9	91,8
Average Accuracy (%)	99,6	99,6
Average Specificity (%)	99,8	99,7
Average time (seconds)	7,6	16,3
Total time (minutes)	5,1	24,2

The used graphical card (GPU) used in the experiments has been an NVIDIA Geforce GT240 composed by 12 *Stream Processors* with a total of 96 cores to 1340 MHz. It has 1 GB DDR3 main memory in a 128 bits bus to 700 MHz. So, the transfer rate obtained is by 54.4 Gbps. The used Constant Memory is 64 KB and the Shared Memory is 16 KB. Its Compute Capability level is 1.2 (from 1.0 to 2.1). The experiments have been performed on a computer with a CPU AMD Athlon II $\times$4 645, which allows to work with four cores of 64 bits to 3.1 GHz. The computer has four blocks of 512 KB of L2 cache memory and 4 GB DDR3 to 1600 MHz of main memory.

All the developed software can be scaled to a more powerful hardware. This represents a new advantage for future developments, since a simple adaptation of the software to new hardware can lead to better results in time.

5. Final remarks

In this paper, a fully automatic computer retinal image analysis system for the localization and segmentation of OD in color digital fundus images has been presented. To this aim, different research work developed by some of the authors in the area of the bioinspired parallel image processing have been put together with some new parallel algorithms implemented for this paper on CUDA^{TM}.

The new implementations developed for this paper have been an improvement gradient operator (color AGP Segmentator), a parallel mask deletion, a new transform for CUDA^{TM} to detect the OD (Hough Circle Cloud) and an improved Hall & Guo algorithm for a parallel thinning. Moreover, we would like to highlight that, on the contrary of other approaches which only detect perfect circles, our algorithm find imperfect circular regions of interest obtaining a better approximation to the OD. Moreover, a previous training with an images group is not needed by our algorithm. So, none time of pre-computation is considered in our case.

We would like to remark that, on the one hand, the proposed algorithm is independent on the color space. This means a conceptual improvement since, if the optical technology changes in the next years, the current parallel implementation of our algorithm will continue obtaining accuracy results in a competitive time. We have created a parallel tool not depending of the input images. On the other hand, the proposed algorithm is completely automatized and no human expert supervision is required in any stage of the process.

Finally, the technique of the cloud HT presented in this paper can be used in further studies related to pathologies or abnormalities in retinal fundus color images.

Table 2
Comparisons with other algorithms for database DRIVE (Top) and DIARETDB1 (Bottom). [(*) Taken from [45]].

Method	Oratio (%)	Sensitivity (%)	Accuracy (%)	Specificity (%)	Performance (%)
Topology cut (*)	55,91	65,12	–	–	–
Adaptive morphologic (*)	41,47	–	–	–	–
Graph cut (*)	55,32	73,98	–	–	–
Graph cut with vad (*)	70,70	84,44	–	–	–
[27]	–	–	95,5	–	100
[18]	–	–	–	–	89
[4]	–	–	–	–	97,5
[46]	–	–	–	–	97,5
Compensation factor (*)	70,9	84,64	–	–	–
MRF image reconstruction (*)	82,4	98,19	–	–	–
Our method	83,3	89,9	99,6	99,8	97,5
Topology cut (*)	38,43	55,30	–	–	–
Adaptive morphologic (*)	43,65	–	–	–	–
Graph cut (*)	54,03	76,35	–	–	–
Graph cut with vad (*)	75,70	86,55	–	–	–
[27]	–	–	93,7	–	97,7
Compensation factor (*)	75,94	86,75	–	–	–
MRF image reconstruction (*)	78,5	87,50	–	–	–
Our method	84,3	91,8	99,6	99,7	97,7

Acknowledgments

MAGN acknowledges the support of the project TIN2012-37434 of the Ministerio de Economía y Competitividad of Spain.

Appendix

Next, we provide a family of tissue P systems to approximate an intensity gradient function of a 2D image. For each image of size n^2 with $n \in \mathbb{N}$, we consider the tissue P system and cell division of degree n^2+2:

$\mathbf{\Pi_2}(r,n) = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, w_{(1,1)}, \ldots, w_{(n,n)}, w_{(1,1)'}, \ldots, w_{(n,n)'} \mathcal{R}, i_{\Pi_2}, o_{\Pi_2})$

where

- $\Gamma = \Sigma \cup \mathcal{E} \cup \{\beta, T, \overline{\gamma}\}$,
- $\Sigma = \{a_{ij} : a \in \mathcal{C},\ 1 \le i,j \le n\}$,
- $\mathcal{E} = \{t_i, (t,1)_i, (t,2)_i, (t,3)_i, \alpha_i : 1 \le i \le n\}$
  $\cup\{a_{ij}, (a,l)_{ij}, : 1 \le i,j \le n,\ 1 \le l \le 6,\ a \in \mathcal{C}\}$
  $\cup\{z_i : 1 \le i \le \beta_1+1,\ \beta_1 = \lceil \log_2 |\mathcal{C}| \rceil\}$
  $\cup\{\delta_i, \overline{\delta}_i, \delta_i', \delta_i'' : 1 \le i \le 4\} \cup \{y_1, y_2, x_1, x_2, x_3\}$
  $\cup\{(A,l)_{ij} : 1 \le i,j \le n,\ 1 \le l \le 4\} \cup \{A_l : 1 \le l \le 4\}$
  $\cup\{p_1, p_2, q_1, q_2, q_3, \mu, \} \cup \{o_l : 1 \le l \le \beta_1+1\}$
- $w_1 = \alpha_1; w_2 = t_1, \ldots, t_n; w_{(1,1)} \cdots = w_{(n,n)} = T, \overline{\gamma}^{255}$,
- $\mathcal{R}$ is the following set of communication rules:
  1. $(1, a_{ij}/(a,1)_{ij}(a,2)_{ij}(a,3)_{ij}, 0)$ for $1 \le i,j \le n$ and $a \in \mathcal{C}$,
  2. $(2, t_i/(t,1)_i, 0)$ for $1 \le i \le n$,
  3. $(1, (a,1)_{ij}(a,2)_{ij}/\lambda, 2)$ for $1 \le i,j \le n$,
  4. $(2, (t,1)_i/(t,2)_i, 0)$ for $1 \le i \le n$,
  5. $(1, \alpha_i/\alpha_{i+1}^2, 0)$ for $1 \le i \le n$,
  6. $(2, (a,1)_{ij}/(a,4)_{ij}^4, 0)$ for $1 \le i,j \le n$,
  7. $(2, (t,2)_i/(t,3)_i, 0)$ for $1 \le i \le n$,
  8. $[(t,3)_i]_2 \to [\beta]_2[\beta]_2$ for $1 \le i \le n$,
  9. $(1, (a,3)_{ij}\alpha_{n+1}/(a,5)_{ij}, 0)$ for $1 \le i,j \le n$,
  10. $(1, (a,5)_{ij}/T, (i,j))$ for $1 \le i,j \le n$,
  11. $((i,j), (a,5)_{ij}/(a,6)_{ij}z_1, 2)$ for $1 \le i,j \le n$,
  12. $((i,j), z_i/z_{i+1}^2, 0)$ for $i = 1, \ldots, \beta_1$,
  13. $((i,j), (a,6)_{ij}/\begin{matrix}(b,2)_{i-1j-1} & (c,2)_{i-1j} & (d,2)_{i-1j+1}\\ (e,2)_{ij-1} & (a,2)_{ij} & (f,2)_{ij+1}\\ (g,2)_{i+1j-1} & (h,2)_{i+1j} & (k,2)_{i+1j+1}\end{matrix}, 2)$
      for $2 \le i,j \le n+1$,
  14. $((i,j), (b,2)_{i-1j-1}(k,2)_{i+1j+1}/\delta_1^b\overline{\delta}_1^k, 0)$ for $1 \le i,j \le n$ and $b,k \in \mathcal{C}$,
  15. $((i,j), (c,2)_{i-1j}(h,2)_{i+1j}/\delta_2^c\overline{\delta}_2^h, 0)$ for $1 \le i,j \le n$ and $c,h \in \mathcal{C}$,
  16. $((i,j), (d,2)_{i-1j+1}(g,2)_{i+1j-1}/\delta_3^d\overline{\delta}_3^g, 0)$ for $1 \le i,j \le n$ and $d,g \in \mathcal{C}$,
  17. $((i,j), (e,2)_{ij-1}(f,2)_{ij+1}/\delta_4^e\overline{\delta}_4^f, 0)$ for $1 \le i,j \le n$ and $e,f \in \mathcal{C}$,
  18. $((i,j), \delta_l\overline{\delta}_l/\lambda, 0)$ for $1 \le i,j \le n$ and $1 \le l \le 4$,
  19. $((i,j), z_{(\beta_1+1)}/y_1 z_{(\beta_1+2)}^4, 0)$ for $1 \le i,j \le n$,
  20. $((i,j), z_{(\beta_1+2)}\delta_l/\delta_l', 0)$ for $1 \le i,j \le n$ and $l = 1,2,3,4$,
  21. $((i,j), z_{(\beta_1+2)}\overline{\delta}_l/\delta_l', 0)$ for $1 \le i,j \le n$ and $l = 1,2,3,4$,
  22. $((i,j), y_1/y_2, 0)$,
  23. $((i,j), \delta_l'\delta_{l+1}'/\lambda, 0)$ for $1 \le i,j \le n$ and $l = 1,3$,
  24. $((i,j), y_2/x_1y_3^2, 0)$ for $1 \le i,j \le n$,
  25. $((i,j), x_l/x_{l+1}, 0)$ for 1, 2 and $l = 1, \ldots, \beta_1+1$,
  26. $((i,j), y_3\delta_l'/\delta_l'', 0)$ for $1 \le i,j \le n$ and $l = 1,2,3,4$,
  27. $((i,j), \delta_l''\delta_k''/\lambda, 0)$ for $1 \le i,j \le n$ and $1 \le l < k \le 4$,
  28. $((i,j), x_3\delta_l''/(A,l)_{ij}, 0)$ for $1 \le i,j \le n$ and $l = 1,2,3,4$,
  29. $((i,j), (A,l)_{ij}/A_l s_1, 0)$
  30. $((i,j), s_l/s_{l+1}^2, 0)$ for $1 \le i,j \le n$ and $l = 1, \ldots, 24\beta_1$,
  31. (a) $((i,j), A_1/\begin{matrix}(a,4)_{i-1j-2} & (b,4)_{i-1j-1}^3\\ (c,4)_{ij-2}^2 & (d,4)_{ij-1}^4\\ (e,4)_{i+1j-2} & (f,4)_{i+1j-1}^3\\ (g,4)_{i-1j+1}^3 & (h,4)_{i-1j+2}\\ (k,4)_{ij+1}^4 & (l,4)_{ij+2}^2\\ (o,4)_{i+1j+1}^3 & (p,4)_{i+1j+2}\end{matrix}, 2)$
      for $2 \le i,j \le n+1$ and $a,b,c,d,e,f,g,h,k,l,o,p \in \mathcal{C}$,
  (b) $((i,j), A_2/\begin{matrix}(a,4)_{i-2j-2}^2 & (b,4)_{i-2j-1} & & \\ (c,4)_{i-1j-2} & (d,4)_{i-1j-1}^4 & (e,4)_{i-1j}^3 & \\ & (f,4)_{ij-1}^3 & & \\ & (g,4)_{ij+1}^3 & & \\ (h,4)_{i+1j}^3 & (k,4)_{i+1j+1}^4 & (l,4)_{i+1j+2}, & 2\\ (o,4)_{i+2j+1}^3 & (p,4)_{i+2j+2}^2 & & \end{matrix})$
      for $2 \le i,j \le n+1$ and $a,b,c,d,e,f,g,h,k,l,o,p \in \mathcal{C}$,
  (c) $((i,j), A_3/\begin{matrix}(a,4)_{i-2j-1} & (b,4)_{i-2j}^2 & (c,4)_{i-2j+1}\\ (d,4)_{i-1j-1}^3 & (e,4)_{i-1j}^4 & (f,4)_{i-1j+1}^3\\ (g,4)_{i+1j-1}^3 & (h,4)_{i+1j}^4 & (k,4)_{i+1j+1}^3\\ (l,4)_{i+2j-1} & (o,4)_{i+2j}^2 & (p,4)_{i+2j+1}, 2\end{matrix})$
      for $2 \le i,j \le n+1$ and $a,b,c,d,e,f,g,h,k,l,o,p \in \mathcal{C}$,
  (d) $((i,j), A_4/\begin{matrix}(a,4)_{i-2j+1} & (b,4)_{i-2j+2}^2\\ (c,4)_{i-1}^3 & (d,4)_{i-1j+1}^4 & (e,4)_{i-1j+2}\\ & (f,4)_{ij+1}^3 & \end{matrix})$

$$\begin{matrix} & & (g,4)^3_{ij-1} & \\ (h,4)_{i+1j-2} & & (k,4)^4_{i+1j-1} & (l,4)^3_{i+1j} \\ (o,4)^2_{i+2j-2} & & (p,4)_{i+2j-1} & \end{matrix}, 2\Bigg)$$

for $2 \leq i, j \leq n+1$ and $a, b, c, d, e, f, g, h, k, l, o, p \in \mathcal{C}$,

32. $((i,j), (z,4)_{ij}/\gamma_1^z, 0)$ for $1 \leq i, j \leq n$ and $z = a, b, c, d, e, f \in \mathcal{C}$,
33. $((i,j), (z,4)_{ij}/\gamma_2^z, 0)$ for $1 \leq i, j \leq n$ and $z = g, h, k, l, o, p \in \mathcal{C}$,
34. $((i,j), \gamma_1\gamma_2/\lambda, 0)$ for $1 \leq i, j \leq n$,
35. $((i,j), s_{24\beta_1+1}\gamma_l\gamma_l/p_1\gamma', 0)$ for $1 \leq i, j \leq n$, $l = 1, 2$,
36. $((i,j), p_1q_1/p_2q_2, 0)$ for $1 \leq i, j \leq n$,
37. $((i,j), \overline{\gamma}\gamma'/\lambda, 0)$ for $1 \leq i, j \leq n$,
38. $((i,j), q_2/q_3, 0)$ for $1 \leq i, j \leq n$,
39. $((i,j), p_2\gamma'/\mu, 0)$ for $1 \leq i, j \leq n$,
40. $((i,j), \mu/(0,6)_{ij}, 0)$ for $1 \leq i, j \leq n$,
41. $((i,j), p_2q_3/q_4, 0)$ for $1 \leq i, j \leq n$,
42. $((i,j), q_4\overline{\gamma}/(1,6)_{ij}o_1, 0)$ for $1 \leq i, j \leq n$,
43. $((i,j), \overline{\gamma}(l,6)_{ij}/(l+1,6)_{ij}, 0)/$ for $1 \leq i, j \leq n$ and $l = 1, \ldots |\mathcal{C}|$,
44. $((i,j), o_l/o_{l+1}, 0)/$ for $1 \leq i, j \leq n$ and $l = 1, \ldots \beta_1$,
45. $((i,j), o_{\beta_1+1}(z,6)_{ij}/a_{ij}, 1)/$ for $1 \leq i, j \leq n$ and $a, z \in \mathcal{C}$,

- $i_{\Pi_2} = 1$
- $i_o = 1$.

References

[1] S. Sekhar, W. Al-Nuaimy, A.K. Nandi, Automated localisation of retinal optic disk using Hough transform, in: Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro. ISBI 2008., IEEE, 2008, pp. 1577–1580.

[2] A. Osareh, M. Mirmehdi, B. Thomas, R. Markham, Automated identification of diabetic retinal exudates in digital colour images., Brit. J. Ophthalmol. 87 (10) (2003) 1220–1223.

[3] D. Kavitha, S. Shenbaga Devi, Automatic detection of optic disc and exudates in retinal images, in: Proceedings of the International Conference on Intelligent Sensing and Information Processing, 2005, 2005, pp. 501–506.

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

[5] M. Madhusudhan, N. Malay, S. Nirmala, D. Samerendra, Image processing techniques for glaucoma detection, in: A. Abraham, J. Mauri, J. Buford, J. Suzuki, S. Thampi (Eds.), Advances in Computing and Communications, Communications in Computer and Information Science, vol. 192, Springer Berlin Heidelberg, 2011, pp. 365–373.

[6] H. Li, O. Chutatape, Boundary detection of optic disk by a modified ASM method, Pattern Recognit. 36 (9) (2003) 2093–2104.

[7] C. Sinthanayothin, J.F. Boyce, H.L. Cook, T.H. Williamson, Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images., Brit. J. Ophthalmol. 83 (8) (1999) 902–910.

[8] M. Langroudi, H. Sadjedi, A new method for automatic detection and diagnosis of retinopathy diseases in colour fundus images based on morphology, in: Proceedings of the International Conference on Bioinformatics and Biomedical Technology (ICBBT), 2010, pp. 134–138.

[9] X. Xu, M. Garvin, M. Abràmoff, J. Reinhardt, Simultaneous automatic detection of optic disc and fovea on fundus photographs, in: SPIE Medical Imaging, International Society for Optics and Photonics, 2011. 79622T–79622T

[10] J. Liu, D. Wong, J. Lim, X. Jia, F. Yin, H. Li, W. Xiong, T. Wong, Optic cup and disk extraction from retinal fundus images for determination of cup-to-disc ratio, in: Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA, 2008, pp. 1828–1832.

[11] X. Xu, M. Niemeijer, Q. Song, M. Sonka, M.K. Garvin, J.M. Reinhardt, M.D. Abràmoff, Vessel boundary delineation on fundus images using graph-based approach, IEEE Trans. Med. Imag. 30 (6) (2011) 1184–1191.

[12] D.K. Wong, J. Liu, J.H. Lim, X. Jia, F. Yin, H. Li, T.Y. Wong, Level-set based automatic cup-to-disc ratio determination using retinal fundus images in ARGALI., Conf. Proc. IEEE Eng. Med. Biol. Soc. 2008 (2008) 2266–2269.

[13] J. Liu, D. Wong, J. Lim, H. Li, N. Tan, Z. Zhang, T. Wong, R. Lavanya, ARGALI: An automatic cup-to-disc ratio measurement system for glaucoma analysis using level-set image processing, in: C. Lim, J. Goh (Eds.), Proceedings of the 13th International Conference on Biomedical Engineering, IFMBE Proceedings, vol. 23, Springer Berlin Heidelberg, 2009, pp. 559–562.

[14] D.K. Wong, J. Liu, J.H. Lim, N.M. Tan, Z. Zhang, S. Lu, H. Li, M.H. Teo, K.L. Chan, T.Y. Wong, Intelligent fusion of cup-to-disc ratio determination methods for glaucoma detection in ARGALI, Conf. Proc. IEEE Eng. Med. Biol. Soc. 1 (2009) 5777–5780.

[15] J. Liu, F.S. Yin, D.W.K. Wong, Z. Zhang, N.M. Tan, C.Y. Cheung, M. Baskaran, T. Aung, T.Y. Wong, Automatic glaucoma diagnosis from fundus image., Conf. Proc. IEEE Eng. Med. Biol. Soc. 2011 (2011) 3383–3386.

[16] G.D. Joshi, J. Sivaswamy, S.R. Krishnadas, Optic disk and cup segmentation from monocular color retinal images for glaucoma assessment, IEEE Trans. Med. Imag. 30 (6) (2011) 1192–1205.

[17] H. Narasimha-Iyer, A. Can, B. Roysam, C.V. Stewart, H.L. Tanenbaum, A. Majerovics, H. Singh, Robust detection and classification of longitudinal changes in color retinal fundus images for monitoring diabetic retinopathy, IEEE Trans. Biomed. Eng. 53 (6) (2006) 1084–1098.

[18] A. Hoover, M.H. Goldbaum, Locating the optical nerve in a retinal image using the fuzzy convergence of the blood vessels, IEEE Trans. Med. Imag. 22 (8) (2003) 951–958.

[19] M. Lalonde, M. Beaulieu, L. Gagnon, Fast and robust optic disc detection using pyramidal decomposition and Hausdorff-based template matching, IEEE Trans. Med. Imag. 20 (11) (2001) 1193–1200.

[20] A.D. Fleming, K.A. Goatman, S. Philip, J.A. Olson, P.F. Sharp, Automatic detection of retinal anatomy to assist diabetic retinopathy screening, Phys. Med. Biol. 52 (2007) 331–345.

[21] K. Akita, H. Kuga, A computer method of understanding ocular fundus images, Pattern Recognit. 15 (6) (1982) 431–443.

[22] A.G. Marrugo, M.S. Millán, Retinal image analysis: preprocessing and feature extraction, J. Phys. Conf. Ser. 274 (1) (2011) 012039.

[23] B. Venkatalakshmi, V. Saravanan, G. Niveditha, Graphical user interface for enhanced retinal image analysis for diagnosing diabetic retinopathy, in: Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011, 2011, pp. 610–613.

[24] B. Harangi, R. Qureshi, A. Csutak, T. Peto, A. Hajdu, Automatic detection of the optic disc using majority voting in a collection of optic disc detectors, in: Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2010, pp. 1329–1332.

[25] M. Park, J.S. Jin, S. Luo, Locating the optic disc in retinal images, in: Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation, IEEE Computer Society, 2006, pp. 141–145.

[26] Z. Liu, C. Opas, S. Krishnan, Automatic image analysis of fundus photograph, in: Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society., 2, 1997, pp. 524–525 vol. 2.

[27] M. Mubbashar, A. Usman, M. Akram, Automated system for macula detection in digital retinal images, in: Proceedings of the International Conference on Information and Communication Technologies (ICICT), 2011, pp. 1–5.

[28] A. Aquino, M. Gegúndez-Arias, D. Marín, Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques, Med. Imag. IEEE Trans. 29 (11) (2010) 1860–1869.

[29] P. Siddalingaswamy, K. Prabhu, Automatic grading of diabetic maculopathy severity levels, in: Proceedings of the International Conference on Systems in Medicine and Biology (ICSMB), IEEE, 2010, pp. 331–334.

[30] C.-Y. Ho, T.-W. Pai, H.-T. Chang, H.-Y. Chen, An atomatic fundus image analysis system for clinical diagnosis of glaucoma, in: Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), IEEE Computer Society, 2011, pp. 559–564.

[31] X. Zhu, R.M. Rangayyan, A.L. Ells, Detection of the optic nerve head in fundus images of the retina using the Hough transform for circles, J. Digital Imag. 23 (2010) 332–341.

[32] M. Niemeijer, M.D. AbrÃ moff, B. van Ginneken, Fast detection of the optic disc and fovea in color fundus photographs, Med. Image Anal. 13 (6) (2009) 859–870.

[33] M.D. Abramoff, W.L.M. Alward, E.C. Greenlee, L. Shuba, C.Y. Kim, J.H. Fingert, Y.H. Kwon, Automated segmentation of the optic disc from stereo color photographs using physiologically plausible features, Invest. Ophthalmol. Vis. Sci. 48 (4) (2007) 1665–1673.

[34] C. Carranza, V. Murray, M. Pattichis, E.S. Barriga, Multiscale AM-FM decompositions with GPU acceleration for diabetic retinopathy screening, in: Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), 2012, pp. 121–124.

[35] S. Moulik, W.W. Boonn, The role of GPU computing in medical image analysis and visualization, in: W.W. Boonn, B.J. Liu (Eds.), Medical Imaging 2011: Advanced PACS-based Imaging Informatics and Therapeutic Applications, vol. 7967, 2011. 79670L–79670L–8, Proceedings of the SPIE.

[36] P.V.C. Hough, Machine analysis of bubble chamber pictures, in: Proceedings of the International Conference on High Energy Accelerators and Instrumentation, CERN, 1959, pp. 554–558.

[37] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures., Commun. ACM 15 (1) (1972) 11–15.

[38] D. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognit. 13 (2) (1981) 111–122.

[39] D. Díaz-Pernil, A. Berciano, F. Peña-Cantillana, M.A. Gutiérrez-Naranjo, Segmenting images with gradient-based edge detection using membrane computing, Pattern Recognit. Lett. 34 (8) (2013) 846–855.

[40] F. Peña-Cantillana, D. Díaz-Pernil, A. Berciano, M.A. Gutiérrez-Naranjo, A parallel implementation of the thresholding problem by using tissue-like P systems, in: P. Real, D. Díaz-Pernil, H. Molina-Abril, A. Berciano, W.G. Kropatsch (Eds.), CAIP (2), Lecture Notes in Computer Science, vol. 6855, Springer, 2011, pp. 277–284.

[41] N. Hamadani, Automatic Target Cueing in IR Imagery, Master's thesis, Air Force Institute of Technology, WAFP, 1981.

[42] F. Peña-Cantillana, A. Berciano, D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, Parallel skeletonizing of digital images by using cellular automata, in: M. Ferri, P. Frosini, C. Landi, A. Cerri, B.D. Fabio (Eds.), CTIC, Lecture Notes in Computer Science, vol. 7309, Springer, 2012, pp. 39–48.

[43] J. Staal, M.D. Abràmoff, M. Niemeijer, M.A. Viergever, B. van Ginneken, Ridge-based vessel segmentation in color images of the retina, IEEE Trans. Med. Imag. 23 (4) (2004) 501–509.

[44] T. Kauppi, V. Kalesnykiene, J.-K. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kälviäinen, J. Pietilä, The DIARETDB1 diabetic retinopathy database and evaluation protocol, in: Proceedings of the British Machine Vision Conference, British Machine Vision Association, 2007, pp. 252–261. University of Warwick, UK, September 10-13, 2007.

[45] A.G. Salazar-Gonzalez, Y. Li, X. Liu, Optic disc segmentation by incorporating blood vessel compensation, in: Proceedings of the IEEE Third International Workshop On Computational Intelligence in Medical Imaging (CIMI), 2011, pp. 1–8.

[46] T. Chaichana, S. Yoowattana, Z. Sun, S. Tangjitkusolmun, S. Sookpotharom, M. Sangworasil, Edge detection of the optic disc in retinal images based on identification of a round shape, in: Proceedings of the International Symposium on Communications and Information Technologies, ISCIT, 2008, pp. 670–674.