

UNIVERSITY OF COIMBRA
FACULTY OF SCIENCES AND TECHNOLOGY
DEPARTMENT OF PHYSICS

BLUEWORKS – MULTIMODAL CLASSIFICATION OF THE RETINA

PEDRO MIGUEL CERVEIRA RODRIGUES

SEPTEMBER, 2010

Project Supervisors: Engineer Paulo Barbeiro
Project Coordinator: Professor Luís Cruz, PhD

A Thesis submitted for obtaining the degree of
Integrated Master in Biomedical Engineering

University of Coimbra
Faculty of Sciences and Technology

ACKNOWLEDGEMENTS

I want to express my thanks to *BlueWorks'* team, particularly to Eng. Paulo Barbeiro, who was very supportive and who provided essential help throughout all year.

Another special thanks to Professor Luís Cruz for the availability and the always helpful insights and orientations.

I would also like to thank Dr. António Travassos, Dr. Eduardo Silva, Dr. Rufino Silva and Dr. Rui Proença for the very helpful responses to the inquiry.

Thanks to Frederico Lopes, João Fernandes and Pedro Sá for the help and all those coffee break brainstorming sessions.

Sincere thanks to Miguel for all the help and friendship.

To my family and my friends.

And finally, to Clau for everything.

ABSTRACT

BlueWorks – Medical Expert Diagnosis has been developing software in the ophthalmology field in order to provide new tools for decision support. This work aims to give further contributions to their projects for multimodal co-registration of ocular fundus images, and to take a step towards a tool for classification of pathologies of the retina and the ocular fundus in general.

In this thesis, it was implemented an algorithm [1] for correction of variability induced in retinal images by non-uniform illumination caused by the image acquisition, as well as some variants to the algorithm's architecture, whose performance and its usefulness for ophthalmologists were subsequently evaluated.

Moreover, modifications were made to the *BlueWorks'* semi-manual image registration project, where ophthalmologists and technicians could perform manual registration to fundus images without much effort, with the help of image processing algorithms and a graphical user interface designed for this purpose. On this line of work, several improvements were made both on the graphical interface level and to the automatic algorithms on which it is based.

In addition, in this document are studied new ways of using SIFT descriptors [2] to register ocular fundus images, combined with approximations [3] at the scale-space analysis level for better speed performance.

RESUMO

A *BlueWorks – Medical Expert Diagnosis* tem vindo a desenvolver software na área da oftalmologia com o intuito de fornecer novas ferramentas para apoio à decisão. Este trabalho tem como objectivo dar continuação aos seus projectos de co-registo multimodal de imagens do fundo ocular e dar um passo na direcção de uma ferramenta de classificação de patologias da retina e do fundo ocular em geral.

Nesta tese foi implementado um algoritmo [1] de rectificação das variações induzidas nas imagens da retina pela iluminação não-homogénea por parte do sistema de aquisição de imagem, assim como algumas variações na arquitectura do algoritmo, cujas performance e a sua utilidade perante oftalmologistas foram posteriormente avaliadas.

Foi também continuado o projecto de registo de imagens semi-manual da *BlueWorks*, em que os oftalmologistas ou técnicos de oftalmologia poderiam registar manualmente as imagens sem muito esforço, com a ajuda de algoritmos de processamento de imagem e uma interface desenhada para este fim. Relativamente a esta linha de trabalho, foram feitas várias melhorias tanto ao nível da interface gráfica como ao nível dos algoritmos automáticos em que esta é fundada.

Para além disso, neste documento são ainda estudadas novas maneiras de usar descritores SIFT [2] para registar imagens do fundo ocular, combinados com aproximações [3] ao nível da análise escala-espço para melhoramento de performance temporal.

CONTENTS

Chapter 1. Introduction.....	1
1.1. Motivation and Objectives.....	1
1.2. Thesis Overview	1
Chapter 2. Ocular Fundus Images	3
2.1. Anatomical Concepts.....	3
2.2. Imaging Systems and Heterogeneity.....	5
Chapter 3. Image Processing.....	8
3.1. Histogram Manipulation Algorithms.....	8
3.1.1. Results	9
3.2. Variability Reduction	9
3.2.1. Brightness and Contrast Normalization	13
3.2.2. Results	18
3.2.3. Discussion.....	22
Chapter 4. Semi-Assisted Alignment.....	24
4.1. Graphical User Interface Improvement	24
4.2. Detection of Y-Features' Center.....	28
4.2.1. Vessel Tracking Algorithm	29
4.3. Discussion.....	31
Chapter 5. Fully-Automated Multimodal Registration.....	32
5.1. Integral Image	33
5.2. Points of Interest Extraction.....	34
5.2.1. Hessian Matrix Determinant Layers.....	34
5.2.2. Accurate Maxima Localization	37
5.3. Orientation Assignment	39

5.4.	SIFT Descriptors and Modifications.....	40
5.4.1.	SIFT Descriptors Review	41
5.4.2.	Architecture I – Modified SIFT Descriptors	42
5.4.3.	Architecture II – Segmentation and SIFT.....	44
5.5.	Descriptor Matching.....	46
5.6.	Transformation.....	47
5.7.	Results	48
5.8.	Discussion.....	52
Chapter 6. Conclusions and Outlook.....		53
Appendix A.	Mask Determination.....	54
Appendix B.	Hessian Matrix.....	55
Appendix C.	Hysteresis Threshold	57
Appendix D.	Otsu’s Threshold.....	59
Bibliography		61

LIST OF FIGURES

FIGURE 1 – DIAGRAM OF THE HUMAN EYE’S ANATOMY. IN WIKIPEDIA [7].....	4
FIGURE 2 – DETAIL OF A COLORED RETINOGRAPHY.	5
FIGURE 3 – THE THREE CHANNELS OF A RGB IMAGE CAPTURED WITH THE <i>TRC-50IX</i> IMAGING SYSTEM. FROM LEFT TO RIGHT: RED CHANNEL, GREEN CHANNEL AND BLUE CHANNEL.....	6
FIGURE 4 – APPLICATION OF THE NEW METHOD OF HISTOGRAM MANIPULATION TO A RGB <i>OPTOMAP</i> ® IMAGE’S HISTOGRAM.	9
FIGURE 5 – REPRESENTATIVE FLOWCHART OF THE ARCHITECTURE USED FOR THE NORMALIZATION OF GRAYSCALE IMAGES....	10
FIGURE 6 – REPRESENTATIVE FLOWCHART OF THE ARCHITECTURE OF APPROACH A USED FOR THE NORMALIZATION OF RGB IMAGES.....	11
FIGURE 7 – REPRESENTATIVE FLOWCHART OF THE ARCHITECTURE OF APPROACHES B AND C USED FOR THE NORMALIZATION OF RGB IMAGES (IMAGES RETRIEVED USING APPROACH C).....	12
FIGURE 8 – RESULTS OF THE APPLICATION OF THE NORMALIZATION TECHNIQUE TO A <i>TRC-50IX</i> RETINOGRAPHY USING THE THREE DIFFERENT IMPLEMENTED APPROACHES (A, B AND C).....	13
FIGURE 9 – REPRESENTATIVE SCHEMES OF THE DEFINITION OF NEIGHBORHOODS ($w/3, h/3$) ON A SQUARE IMAGE.....	16
FIGURE 10 – BACKGROUND ESTIMATION USING THE SLOWEST APPROACH.	16
FIGURE 11 – ILLUSTRATION OF THE DRIFTS COMPUTED FROM THE GRAYSCALE IMAGE SHOWN IN FIGURE 10 WITH THE SLOWEST APPROACH. LEFT: CONTRAST DRIFT. RIGHT: BRIGHTNESS DRIFT.	17
FIGURE 12 – COMPARATIVE CHART CONTAINING THE BACKGROUND MAP APPROXIMATION ERRORS FOR BOTH THE SECOND APPROACH AND THE ONE WITH OVERLAP IN RELATION TO THE SLOWER APPROACH.	19
FIGURE 13 – COMPARATIVE CHART CONTAINING THE BRIGHTNESS DRIFT APPROXIMATION ERRORS FOR BOTH THE SECOND APPROACH AND THE ONE WITH OVERLAP IN RELATION TO THE SLOWER APPROACH.	20
FIGURE 14 – COMPARATIVE CHART CONTAINING THE CONTRAST DRIFT APPROXIMATION ERRORS FOR BOTH THE SECOND APPROACH AND THE ONE WITH OVERLAP IN RELATION TO THE SLOWER APPROACH.	20
FIGURE 15 – COMPARATIVE CHART CONTAINING THE OUTPUT IMAGE APPROXIMATION ERRORS FOR BOTH THE SECOND APPROACH AND THE ONE WITH OVERLAP IN RELATION TO THE SLOWER APPROACH.	21
FIGURE 16 – CHART CONTAINING THE OPHTHALMOLOGISTS’ RESPONSES.....	22
FIGURE 17 – SCREENSHOT OF THE FORMER GRAPHICAL USER INTERFACE.	25

FIGURE 18 – DETAIL OF BOTH THE PREVIOUS AND NEW INTERFACES.	26
FIGURE 19 – DETAIL OF THE FORMER INTERFACE SHOWING THE AREA WHERE THE POINT MATCHING IS PERFORMED. IN LOPES [13].	26
FIGURE 20 – SCREENSHOT OF THE NEW GRAPHICAL USER INTERFACE.	27
FIGURE 21 – DETAIL OF THE NEW INTERFACE SHOWING THE AREA WHERE THE POINT MATCHING IS PERFORMED.	27
FIGURE 22 – PRE-PROCESSING STEP FOR THE Y-FEATURES’ CENTER DETECTION ALGORITHM.	28
FIGURE 23 – GENERIC MODEL OF THE VESSEL PROFILE USED IN THE TRACKING ALGORITHM.	30
FIGURE 24 – REPRESENTATIVE FLOWCHART OF THE REGISTRATION ALGORITHMS IMPLEMENTED.	32
FIGURE 25 – DIAGRAM OF A GENERIC RECTANGULAR REGION IN AN UPRIGHT POSITION.	34
FIGURE 26 – TEMPLATES FOR COMPUTING THE IMAGE’S INTENSITY SECOND-ORDER DERIVATIVES. IN EVANS [18].	35
FIGURE 27 – SCHEME SHOWING A TYPICAL PYRAMIDAL PROCESS (LEFT) AND THE USED PYRAMIDAL PROCESS (RIGHT).	36
FIGURE 28 – REPRESENTATIVE SCHEME OF THE DISTRIBUTION OF THE KERNEL’S SIZE ACROSS THE OCTAVES AND THEIR RELATION WITH THE SCALE THEY ARE INTENDED TO.	37
FIGURE 29 – ILLUSTRATION OF THE SEARCH FOR LOCAL MAXIMA IN A GENERIC POINT (RED MARK) BY COMPARING IT TO ITS IMMEDIATE NEIGHBORS (BLUE SQUARES). ADAPTED FROM [2].	38
FIGURE 30 – ORIENTATION ASSIGNMENT ILLUSTRATION SHOWING THE RESPONSE VECTORS (BLUE POINTS), THE ANGLE APERTURE BEING ANALYSED (DARKER REGION) AND THE ASSIGNED ORIENTATION (RED VECTOR). IN BAY ET AL. [3]. ...	39
FIGURE 31 – HAAR WAVELET FILTERS WITH BLACK AND WHITE REGIONS WEIGHTED WITH -1 AND +1, RESPECTIVELY.	39
FIGURE 32 – ROTATED GRADIENT SAMPLES IN AN 8X8 QUADRANGULAR REGION AROUND A KEY POINT AND THEIR DISTRIBUTION IN THE CORRESPONDING 2X2 HISTOGRAMS. IN LOWE [2].	41
FIGURE 33 – ILLUSTRATION OF THE COMPUTATION OF A SYMMETRIC DESCRIPTOR. IN CHEN ET AL. [19].	43
FIGURE 34 – THE TWO TEMPLATE TYPES THAT ARE USED TO PRE-PROCESS OCULAR FUNDUS IMAGES. ADAPTED FROM LOPES [13].	45
FIGURE 35 – CHART CONTAINING THE NUMBER OF MATCHES USING ARCHITECTURE I (SYMMETRIC DESCRIPTORS)	49
FIGURE 36 – EXAMPLE OF THE IMAGES CREATED FROM THE DRIVE DATABASE.	50
FIGURE 37 – CHART CONTAINING THE NUMBER OF MATCHES USING ARCHITECTURE II ON THE DATA SET WITH PERSPECTIVE TRANSFORMATION.	51
FIGURE 38 – CHART CONTAINING THE NUMBER OF MATCHES USING ARCHITECTURE II ON THE DATA SET WITH PERSPECTIVE TRANSFORMATION AND VESSEL DELETION.	51
FIGURE 39 – MASK DETERMINATION ILLUSTRATION.	54
FIGURE 40 – REPRESENTATIVE DIAGRAM OF THE APPLICATION OF HYSTERESIS THRESHOLDING TO A SET OF INTESITIES ALONG A ROW OF THE IMAGE. THE RED LINES REPRESENT THE TWO TRESHOLDS, THE BLUE PART ARE THE PIXELS CLASSIFIED AS “TRUE” AND THE BLACK PART OF THE CURVE ARE THE PIXELS CLASSIFIED AS FALSE.	57

LIST OF ACRONYMS

DLL	Dynamic-Link Library
GUI	Graphical User Interface
HHFW	Half Height at Full Width
HSV	Hue, Saturation, Value (color space)
RGB	Red, Green, Blue (color space)
ROI	Region of Interest
SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Features

CHAPTER 1. INTRODUCTION

1.1. MOTIVATION AND OBJECTIVES

The main objective of this project is to contribute to the creation of an analysis system for multimodal ocular fundus images that will allow visualization, registration and pathology progress evaluation and classification on such images. The most crucial step in this kind of system is the registration process, since without it there can be no pathology progression evaluation. Moreover, a reliable mechanism that performs fully-automated registration of retinal images is also an extremely valuable tool for any ophthalmologist, because it allows them to have a very easy way to cross information between exams from different modalities, different stages of pathology progression or even from the same exam set but different retinal areas (for instance, when the area of interest is not entirely covered by a single image). Therefore, all implementations and decisions that were made during this project were made with the future of such system in mind.

1.2. THESIS OVERVIEW

This document is segmented in four main chapters. The first of the four – chapter 2 – is meant to provide an overview of the eye’s anatomy, along with an introduction to the heterogeneity problem and to the characteristics of ocular fundus images. The imaging systems and the exam types used throughout this project are also presented in that chapter.

Chapter 3 describes an algorithm to normalize image parameters, which follows the proposal of Foracchia *et al.* [1], as well as some variations to its architecture. All implemented

architectures and approaches to speed-up the process of normalizing the image are then extensively examined and other approaches are discussed.

This algorithm was implemented in order to address part of the heterogeneity problem that exists in this type of images, so that, other automatic algorithms of pathology progression evaluation and retinal classification in general can be used. Therefore, this study aims to contribute to multimodal system that, in the future, is meant to automatically perform pathology progression assessment from registered images of different imaging systems. However, for such algorithm to be designed and implemented, one needs to have an image alignment algorithm. Chapters 4 and 5 aim to present alternatives to the co-registration step.

The work described in chapter 4 is a continuity project of a previous *Blueworks* semi-assisted solution to perform registration of two fundus images. Several improvements to both the graphical user interface and automated part of the software were made. However, this line of work was suspended by the Project supervisor. This was done due to the results obtained while searching for procedures to optimize the automated identification of vascular points, since it was identified that some of them described the ability to properly solve the more complex problem of fully-automated alignment. Acknowledging that the implementation tasks requiring the most effort were common to both full-automated and semi-automated approaches, as well as the potential outcome, it was decided that there was a positive cost-benefit ratio to pursue the more complex registration and to study other ways of performing fully-automated registration.

To conclude, chapter 5 describes two fast architectures of modified SIFT descriptors that were used to try to perform multimodal registration of fundus images.

CHAPTER 2. OCULAR FUNDUS IMAGES

2.1. ANATOMICAL CONCEPTS

The human eye is composed by the eyeball and its protective structure. The eyeball is a spherical organ with a diameter of 2,5 cm and it is organized in three chambers: the anterior chamber, the posterior chamber and the vitreous chamber. The anterior chamber is positioned between the cornea and the iris and communicates with the posterior chamber, which is localized between the iris and the lens. Both of them are filled with a liquid substance called aqueous humour. The vitreous chamber is the biggest cavity of human eye and contains the vitreous humour, a gel composed mainly by water [4]. [5]

The eyeball's wall is composed by three layers: the outer layer, known as the fibrous tunic, the choroid, in the middle, and the inner layer, which is called retina. The first contains the cornea, a transparent layer localized in the anterior pole of the eyeball, and the sclera, the white thick layer that externally covers the rest of the eye. The middle layer, or the choroid, is composed by the ciliary body, the iris and the choroidea. The choroidea is a highly vascularized structure composed mainly by collagen and elastic fibers, capillary and an external layer with medium-size vessels and areolar connective tissue. Lastly, the inner layer of the eyeball's wall, the retina, comprises two parts: a photoreceptive part (*pars optica retinae*) and a nonreceptive part (*pars caeca retinae*) that merge together at the *ora serrata*. The photoreceptive part is organized in several layers composed mainly by neural cells. The outer layers of the retina are supplied by the capillary net of the choroid, whereas the inner layers receive their supply from the central artery of retina. [5, 6]

The optic disk, situated at the posterior pole of the eyeball, is the entry point of the central retinal vessels and the optic nerve.

Figure 1 depicts the human eye's anatomy.

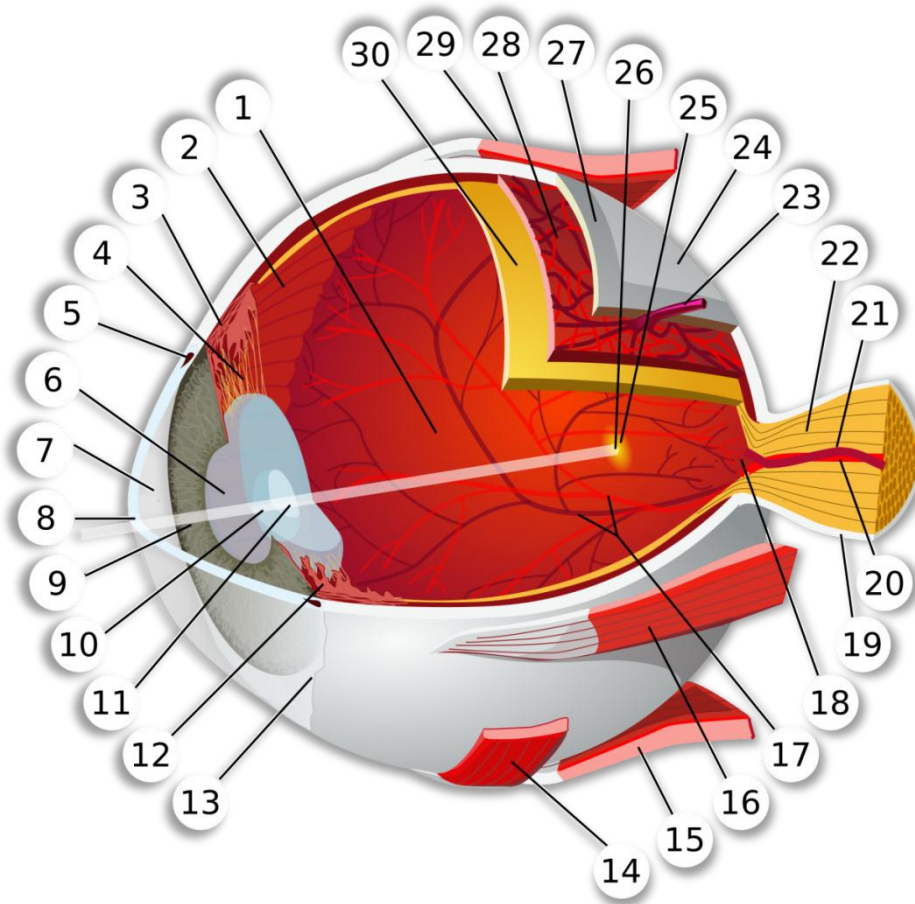


Figure 1 – Diagram of the human eye’s anatomy. In Wikipedia [7].

Key: 1. Vitreous Chamber; 2. Ora Serrata; 3. Ciliary Muscle; 4. Ciliary Zonules; 5. Canal Of Schlemm; 6. Pupil;
 7. Anterior Chamber; 8. Cornea; 9. Iris; 10. Lens Cortex; 11. Lens Nucleus; 12. Ciliary Process; 13. Conjunctiva;
 14. Inferior Oblique Muscle; 15. Inferior Rectus Muscle; 16. Medial Rectus Muscle; 17. Retinal Arteries And Veins;
 18. Optic Disc; 19. Dura Mater; 20. Central Retinal Artery; 21. Central Retinal Vein; 22. Optic Nerve;
 23. Vorticosse Vein; 24. Bulbar Sheath; 25. Macula; 26. Fovea; 27. Sclera; 28. Choroid; 29. Superior Rectus Muscle;
 30. Retina.

The ocular fundus, extensively referred to throughout this thesis, is formed by the inner structures of the posterior pole of the eyeball, namely, the retina, optic disk, central vasculature of the retina, macula, fovea and the choroid (see Figure 2). Noteworthy is the fact that a normal retina is colorless and the red coloration seen in colored retinographies is inflicted by the choroid’s vascular net [6, 8].

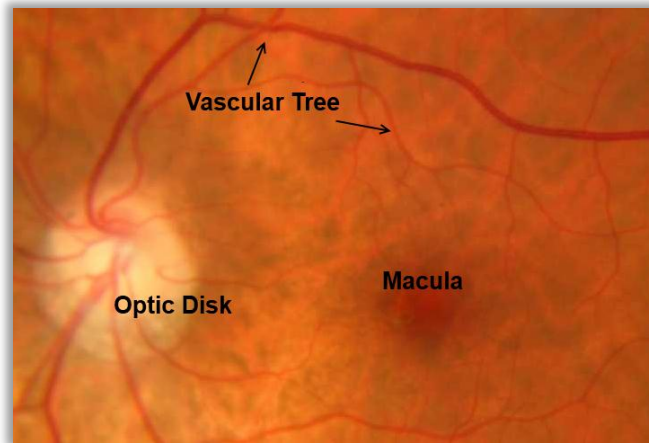


Figure 2 – Detail of a colored retinography.

2.2. IMAGING SYSTEMS AND HETEROGENEITY

This project is addressed to ocular fundus images in general, which poses a significant problem, since there is huge variability in these images, both within each image (intra-variability) and between images (inter-variability). This heterogeneity has its source in:

- The large number of acquisition methods and imaging systems
- The high number of pathologies registered by these methods
- Non-uniform illumination
- The inter-patient heterogeneity inflicted by age, race and the personal features of each patient

The algorithms discussed throughout this thesis were implemented to target images acquired with several imaging systems and methods of acquisition. Table 1 shows the main image types that were used in the testing and to generate results.

Table 1 – Acquisition systems and exam types used in this project. Imaging systems: *Heidelberg Engineering Spectralis® HRA and HRA2, TOPCON TRC-50IX and TRC-NW8 and Optos® optomap®*.

Exam Types	Imaging Systems			
	<i>HRA/HRA 2</i>	<i>TRC-50IX</i>	<i>TRC-NW8</i>	<i>optomap®</i>
RGB Retinographies	-	x	x	x ¹
Red-free Retinographies	x	x	x	x
Angiography with Fluorescein	x	x	-	x
Auto-flourecence Imaging	x	-	-	-
Infra-Red	x	-	-	-

A typical RGB (red, green and blue) fundus image contains very different information in each channel (see Figure 3). The blue channel normally shows the optic disk and some pathological structures, but, normally, it does not have very useful information. Moreover, as mentioned before, the red coloration of fundus images is generated by the reflection of light on the vasculature of the choroid. Therefore, the red channel contains mainly information on the choroid level.

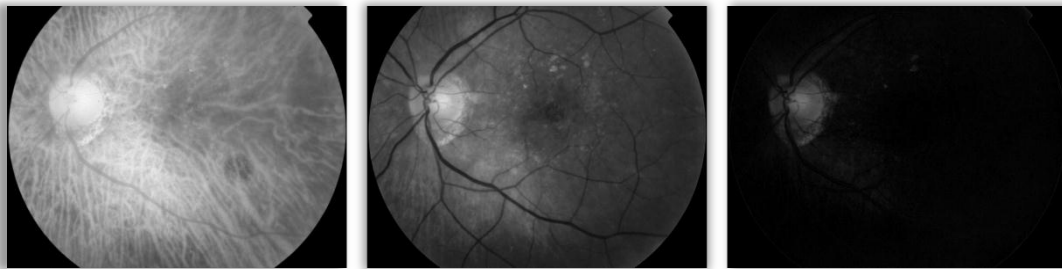


Figure 3 – The three channels of a RGB image captured with the *TRC-50IX* imaging system. From Left to right: red channel, green channel and blue channel.

As regards to the green channel, that is where the most interesting information can be observed. In fact, “empirical observations by several authors (*e.g. Shin et al.*²; Leandro *et al.*³) identify the green channel of RGB images as containing the maximum contrast. Rapantzikos *et*

¹ In optmap®’s colored retinographies the blue channel is approximately zero for all pixels. However, in the implementations discussed in this thesis, these images are always treated as normal RGB images.

² Shin DS, Javornik NB, Berger JW. *Computer-assisted, interactive fundusimage processing for macular drusen quantitation*. *Ophthalmology* 1999;106:1119–25.

³ Leandro JJG, Soares JVB, Cesar Jr RM, Jelinek HF. *Blood vessels segmentation in nonmydriatic images using wavelets and statistical classifiers*. In: XVI Brazilian symposium on computer graphics and image processing. 2003. p.262–9.

al.⁴ also note that the green channel appears to provide “more information” and is less subject to non-uniform illumination” [9]. Therefore, some algorithms like vessel segmentation only use the green channel instead of the whole image or a grayscale version of it, since the vasculature and other structures are better differentiated from the background in this channel.

Another relevant feature of this type of images is the black mask that appears on some images. This can be seen in Figure 3 and is caused by a shadow of the acquisition system. Most algorithms have to compensate for this effect and, therefore, ways of detecting it had to be used.

⁴ Rapantzikos K, Zervakis M, Balas K. *Detection and segmentation of drusen deposits on human retina: potential in the diagnosis of age-related maculardegeneration*. Med Imaging Anal 2003;7:95–108.

CHAPTER 3. IMAGE PROCESSING

3.1. HISTOGRAM MANIPULATION ALGORITHMS

Medical images of the ocular fundus come in a wide range of sizes, going from images with $3,1 \times 10^5$ pixels to images with almost $1,2 \times 10^7$ pixels that can either have one or three color planes. Therefore, when performing even trivial tasks, like brightness manipulation, in larger images, computing time becomes a critical factor.

In order to replace the algorithms implemented in Visual C#, already integrated in the *BlueWorks* software, new methods were implemented to manipulate brightness and contrast, along with an algorithm to perform gamma correction and a new histogram manipulation algorithm inspired by *optomap*[®] images' histograms.

The histograms of retinographies captured with the *optomap*[®] imaging system have, in general, intensity values comprised in the lower values of the histogram's spectrum in all three color planes (see Figure 4). This fact brought to mind an algorithm that is potentially useful to ophthalmologists and technicians, particularly in the examination of such images. The implemented algorithm uses a threshold value between the spectrum boundaries (*i.e.*, between 0 and 255) given by the user through the GUI, to crop the pixels with values below that threshold – eliminating all pixels with intensity values in the red zone (Figure 4) – and stretch the new cropped histogram to fit the spectrum boundaries.

This will increase the contrast and consequently highlight several structures losing only a small percentage in the higher values of the histogram.

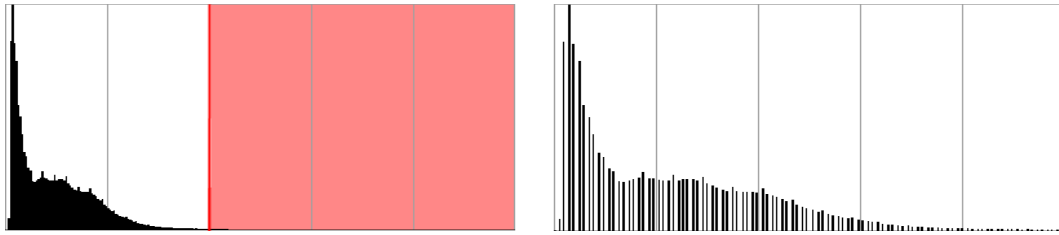


Figure 4 – Application of the new method of histogram manipulation to a RGB *optomap*[®] image's histogram.
Left: original histogram marked with the cutting point. Right: output image's histogram.

To achieve high computation performance the methods for computing the manipulated images were implemented in Visual C++ and compiled into a DLL file meant to be used in the *BlueWorks* software, which is a Visual C# project. All mentioned algorithms were implemented with this architecture in mind, allowing an optimization of the interaction between languages.

3.1.1. RESULTS

There were massive improvements on the timings of these manipulation algorithms. The processing speed was reduced to approximately 2,5% of the older algorithms' processing time for RGB images and reaching 2,1% of the previous processing time on grayscale images. These time measurements were obtained with the application of brightness, contrast and gamma correction simultaneously to images with dimensions of 3900 by 3072 pixels.

3.2. VARIABILITY REDUCTION

To address the intra-image variability issue, already introduced in chapter 2, an algorithm based on the method proposed by Foracchia *et al.* [10] was implemented. This method aims to compensate for the variability in the image's contrast and brightness⁵ caused by non-uniform illumination, which can affect the diagnosis by rendering some lesions and pathologies barely visible to a human observer. Furthermore, intra-image variability reduction may also be

⁵ The term used by Foracchia *et al.* in the same context is luminosity, which is a measurement of brightness. However, for coherence purposes, brightness is going to be used in this thesis.

essential to other algorithms like region segmentation or pathology progression evaluation algorithms, depending on their approaches to solve the respective problems.

This method consists in computing the brightness and contrast drifts, using the image's background information, to correct illumination variability. Note that background in this context is thought as the image itself without any visible pathological structures or vasculature [10].

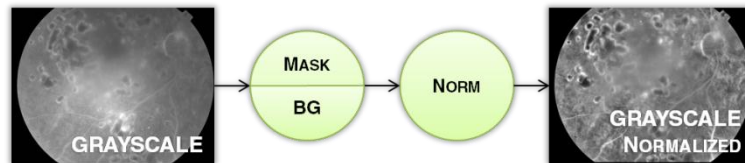


Figure 5 – Representative flowchart of the architecture used for the normalization of grayscale images.

Key: Mask – mask determination; BG – background estimation; Norm – image normalization and post-processing.

Regarding grayscale images, such as angiographies and red-free retinographies, the overall architecture of the algorithm is very straightforward (see Figure 5). The source image is used to perform mask determination and background estimation and then the normalization process is applied to image generating the corrected grayscale image. However, for three-channel RGB images, further attention must be paid to the architecture of the algorithm, since there are 3 channels with very different information, as shown in chapter 2. If the purpose of the algorithm is to normalize RGB images to perform segmentation of the retinal vascular tree, then the red and blue channels are of little use and can be discarded [9], and the method should simply be applied to the green channel. On the other hand, if the objective is the normalization for better visualization or to use as a pre-processing step before the application of other algorithms that deal with color information like pathology detection and evaluation algorithms, then the three channels must be considered and processed so that no color information is lost. To address this situation, three different approaches to this problem were implemented and studied. For now they are going to be referred as approaches A, B and C.

Approach A follows an architecture similar to one mentioned by Bernardes⁶ in the same context. In this approach the source image is transformed into two different color spaces, namely HSV (hue, saturation and value) and grayscale (see Figure 6). The grayscale version of

⁶ In a lecture entitled *Imagiologia do Fundo Ocular: do pixel à função* by Professor Rui Bernardes on 10/03/2009, as part of a lecture series sponsored by NEDF/AAC. Coimbra, Portugal.

the image loses hue and saturation information, which, however, can be restored using the HSV image. Therefore, all processing steps are applied only to the grayscale image and then the processed grayscale image is combined with the hue and saturation channels to generate the final RGB normalized image. By doing this it is being assumed that the grayscale image is equivalent to the value channel ($V = \max(R, G, B)$), *i.e.*, each pixel of the grayscale image is equivalent to maximum of the corresponding three pixels from the red, green and blue channels. This is not true, however, it is not a bad assumption to be made since all important features can be observed in the grayscale image and, therefore, the pixels in the grayscale can be considered of being of more importance than the actual maximum.

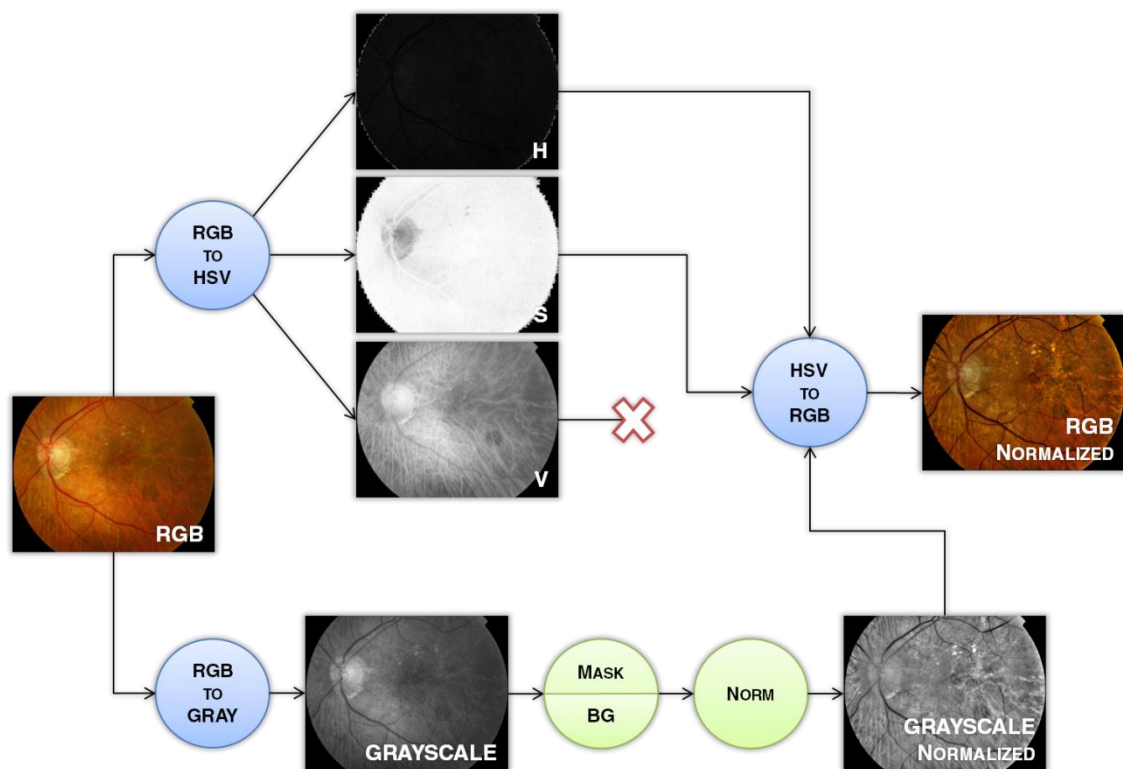


Figure 6 – Representative flowchart of the architecture of approach A used for the normalization of RGB images. Key: Mask – mask determination; BG – background estimation; Norm – image normalization and post-processing.

The architecture used in approaches B and C (see Figure 7) is based in the aforementioned paper by Foracchia *et al.* [10], in which it is proposed the application of the algorithm to the three channels independently. Nevertheless, the architecture was slightly changed mainly to shorten the processing time. In these approaches, the brightness and contrast normalization are still applied to the channels in parallel, but the background estimation is applied only to a grayscale version of the source image, instead of being performed for each channel. This way

the background is estimated according to the combination of all channels and the same binary map of the background is used in the normalization processes of the three channels.

Furthermore, the difference between approaches B and C is not on the architecture *per se* but in a step of post-processing which will be discussed later.

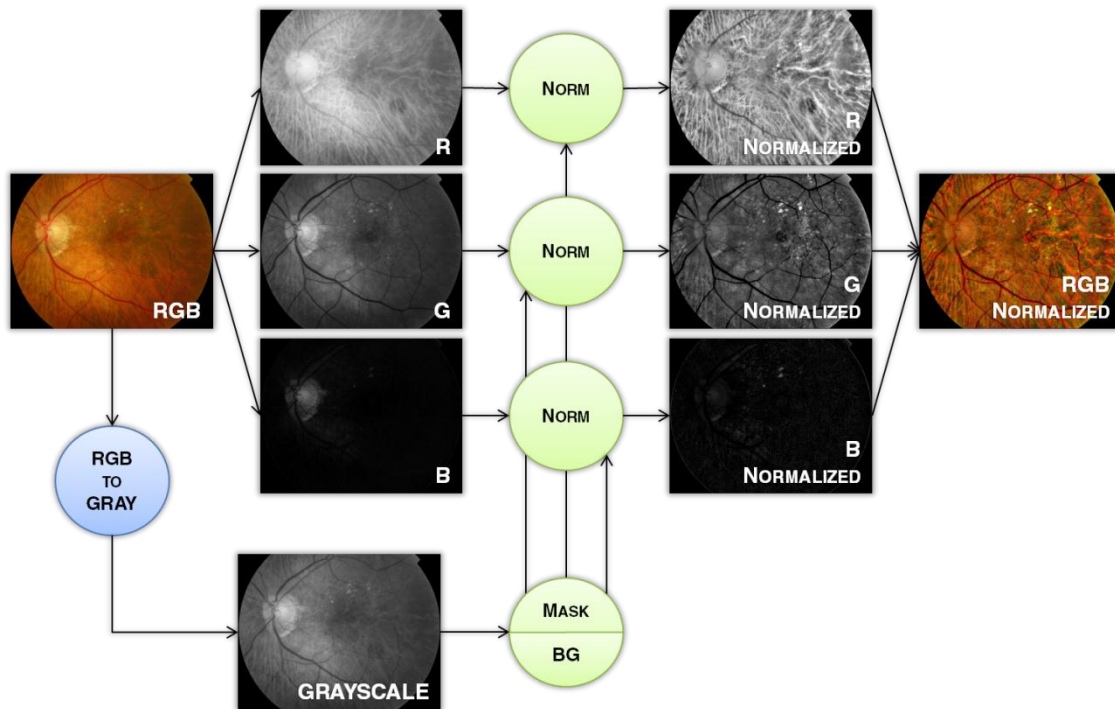


Figure 7 – Representative flowchart of the architecture of approaches B and C used for the normalization of RGB images (images retrieved using approach C).

Key: Mask – mask determination; BG – background estimation; Norm – image normalization and post-processing.

An example of the application of the three approaches is shown in Figure 8.

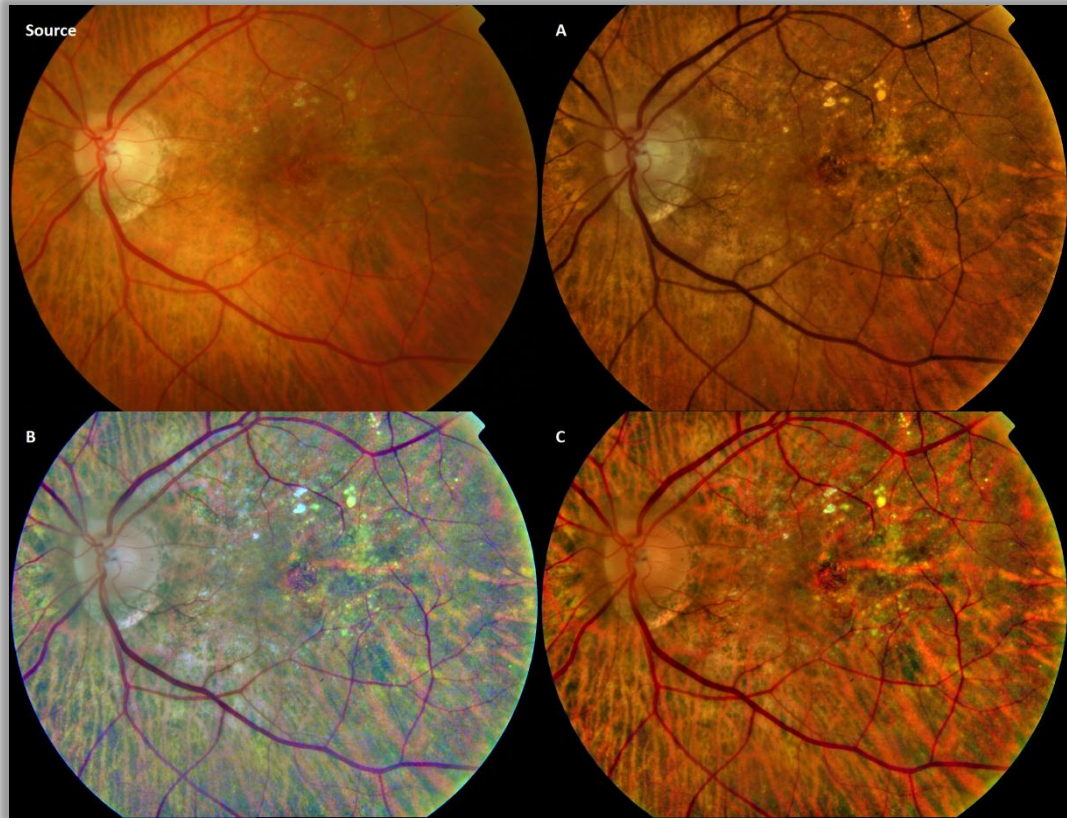


Figure 8 – Results of the application of the normalization technique to a *TRC-50IX* retinography using the three different implemented approaches (A, B and C).

3.2.1. BRIGHTNESS AND CONTRAST NORMALIZATION

The brightness and contrast normalization technique proposed by Foracchia *et al.* [10] is based on a simple model where the observed image of the ocular fundus (I_{ob}) or, in other words, the image retrieved from the imaging system, is thought as a function (f) of the actual image or the original image (I_{or}):

$$I_{ob}(x, y) = f(I_{or}(x, y)) \quad (3.1)$$

Therefore, if that function, which characterizes the acquisition system, was known and invertible, one could reconstruct the original image. However, that function can only be predicted to a certain level of accuracy and, since foreground properties are rather difficult to predict and to model, to estimate f only background pixels are used.

Considering that the ideal image's background information can be modeled as a normal distribution with mean μ_{bg} and standard deviation σ_{bg} , that correspond to the overall brightness and contrast values, respectively, as long as one only aims to compensate for non-uniform illumination, in order to estimate I_{or} , one only needs to find the drifts of this values in the observed image. In other words, the acquisition function could be described with the drifts computed by local means and local standard deviations of the background of I_{ob} that correspond to the contrast (C) and the brightness (B) drifts, respectively. Therefore, the acquisition function becomes:

$$f(I_{or}(x, y)) = C(x, y)I_{or}(x, y) + B(x, y) \quad (3.2)$$

Note that C and B are matrices with the same size as the images.

Mask determination is the first step of the normalization algorithm and it is done with an adaptation of the implementation of Chanwimaluang *et al.* [11] which is described in Appendix A.

For better understanding of the rest of the algorithm, it will be divided into two main steps:

- Background estimation
- Image normalization and post-processing

BACKGROUND ESTIMATION

In this step, for each pixel a neighborhood is defined and then the mean and standard deviation of the values of the pixels belonging to this neighborhood are computed, resulting in two matrices with the same size as the image, μ_N and σ_N , respectively. That information is then used to categorize the pixels in two distinct sets, foreground and background. This classification is done by computing for each pixel the Mahalanobis distance according to the following:

$$d_{Mahalanobis}(x, y) = \left| \frac{I_{ob}(x, y) - \mu_N(x, y)}{\sigma_N(x, y)} \right| \quad (3.3)$$

and then by comparison with a threshold (t_M) each pixel is assigned to one of two categories giving origin to a binary map of the background, *i.e.*, those pixels that satisfy the condition $d_{Mahalanobis} \leq t_M$ are categorized as background; those that do not are set as belonging to the foreground. The threshold (t_M) was set to 1, as proposed by Foracchia *et al.* [10], meaning

that if the intensity value of the pixel with coordinates (x, y) belongs to the interval $[\mu_N(x, y) - \sigma_N(x, y), \mu_N(x, y) + \sigma_N(x, y)]$ it will be marked as background, otherwise it is assigned as foreground.

Pixels that belong to the mask do not enter in the computation of either μ_N or σ_N and they are set as foreground.

For the background estimation procedure, some assumptions regarding the neighborhoods were made [10] that must be considered while choosing its size:

1. Both B and C are constant within the neighborhood
2. At least 50% of the pixels in the neighborhood belong to the background
3. Intensities of the background pixels are significantly different from those of the foreground pixels

Therefore, the size of the neighborhoods must be carefully chosen, since it should obey the second assumption and still have no significant variations in contrast and luminosity, as stated in the first assumption. The size used for the neighborhoods can be adjusted, however, by default, it was chosen to be $\frac{1}{8}$ of the size of the image, similar to the size used in the paper [10]. In other words, the neighborhood is a matrix $N_{\frac{w}{8} \times \frac{h}{8}}$, where w and h are the image's width and height, respectively.

The aforementioned approach is very expensive from a processing standpoint, which makes it not viable in practice. Thus, a similar but much faster way to estimate the background was also stated in the same paper [10] and also implemented in this project. In this implementation, instead of computing μ_N and σ_N for each pixel of the observed image, the pixels of the image are divided into a tessellation of neighborhoods. The implementation made in this project follows the scheme depicted on the left of Figure 9. The neighborhoods, by default, have approximately the same size as before, which results in 81 $((8 + 1)^2)$ neighborhoods and the same number of means and standard deviations. The matrices μ_N and σ_N are then found by assigning those values to the central pixel of each neighborhood and using a bicubic interpolation to compute the remaining values. This is acceptable since B and C should be approximately constant within each neighborhood.

Another approach was also implemented that, in practice, takes almost the same time to execute. This method uses neighborhoods with the same size, but with approximately 50% overlap between them (see scheme on the left of Figure 9). Resulting in 289 $((2 * 8 + 1)^2)$ regions and, consequently, a more accurate interpolation.

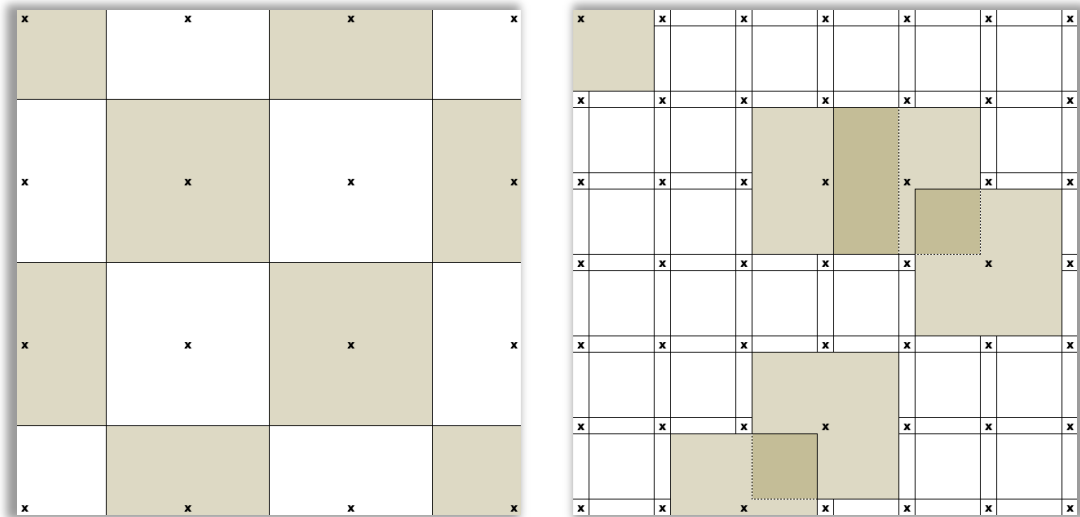


Figure 9 – Representative schemes of the definition of neighborhoods ($w/3, h/3$) on a square image.
 Left: tessellation of neighborhoods used in the second approach. Right: neighborhoods with overlapping.

An example of the binary map of the background's estimation using the slower approach is shown below, in Figure 10.

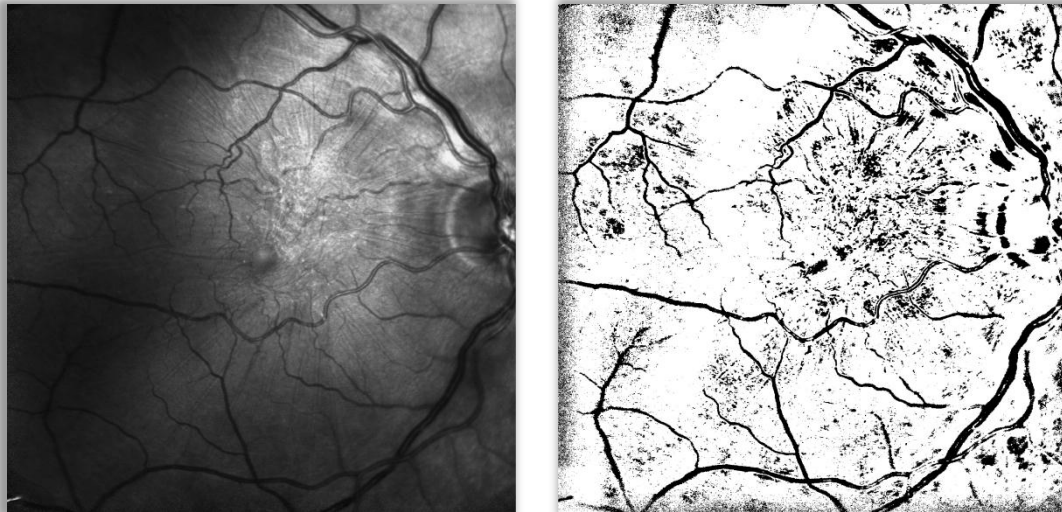


Figure 10 – Background estimation using the slowest approach.
 Left: Observed image. Right: binary map of the background.

Note that the background classification does not need to have high sensitivity, allowing a much faster approach than the ones usually used for region segmentation in this type of images. It should, however, have a relatively high specificity, meaning that a pixel classified as

background should belong with a high level of certainty to the background. Otherwise, pixels wrongly classified as background would be incorporated in the drifts' computations, causing the modulation as a normal distribution to be incorrect.

IMAGE NORMALIZATION AND POST-PROCESSING

The next step is the estimation of background drifts. This stage is very similar to the previous one, with the exception that the means and standard deviations are now computed with only background pixels, resulting, as mentioned before, in two matrices (\hat{C} and \hat{B}), with the same size as the observed image, that describe the brightness and contrast variations throughout the observed image.

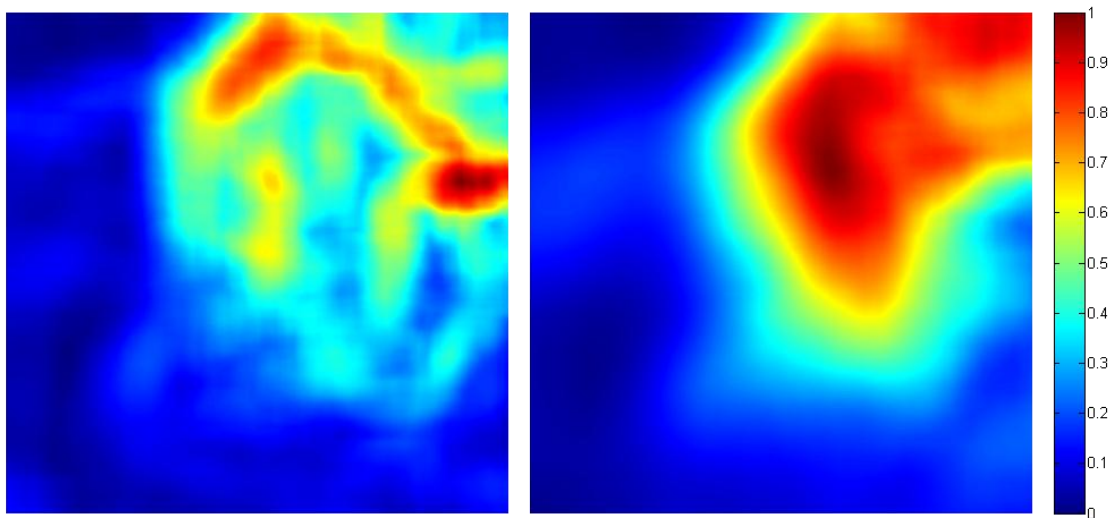


Figure 11 – Illustration of the drifts computed from the grayscale image shown in Figure 10 with the slowest approach. Left: contrast drift. Right: brightness drift.

All three approaches to the definition of the neighborhoods, used in the previous stage, were also implemented in this step. Namely, the two approaches introduced by Foracchia *et al.* [10] and the one with overlapping.

After obtaining the drifts, the observed image is corrected to approximate the ideal retinal image, free of illumination variability. This is done simply by combining equations 3.1 and 3.2, resulting in:

$$\hat{I}_{or}(x, y) = \frac{I_{ob}(x, y) - \hat{B}(x, y)}{\hat{C}(x, y)} \quad (3.4)$$

where \hat{I}_{or} is the estimate of the actual fundus image.

In practice, the corrected image will have some intensity outliers, either because of localized noise in the observed image or because of an over-enhancement of image features associated with pathology induced structural change. These outliers are eliminated in post-processing by considering only the values that fall inside the interval $[\mu - 5\sigma, \mu + 5\sigma]$ and assigning the remaining values with the boundaries values, according to the equation below.

$$I'(x, y) = \begin{cases} \mu - 5\sigma, & I(x, y) < \mu - 5\sigma \\ I(x, y), & \mu - 5\sigma \leq I(x, y) \leq \mu + 5\sigma \\ \mu + 5\sigma, & I(x, y) > \mu + 5\sigma \end{cases} \quad (3.5)$$

This way, the outliers are eliminated without eliminating any relevant information. The image's histogram is then renormalized to fit the range of 8bit images ($[0, 255]$). Grayscale normalization and approaches A and B use this method to eliminate outliers. However, this is where approaches B and C diverge, since in the latter the overall mean and standard deviation of I_{ob} (excluding pixels that belong to the mask) are restored to \hat{I}_{or} . Therefore the result of approach C may lose more information in each channel, but also holds color information that it is lost in approach B.

3.2.2. RESULTS

All the following numbers and charts were obtained using the default value for the size of the neighborhoods ($\frac{1}{8}$ the input image's size).

To analyze the speed of the three approaches used in the definition of the neighborhoods, a benchmark was created to use a grayscale image with 1320 by 1030 pixels to compute the mean processing time for each approach throughout several repetitions of the same processing task. The results obtained were the following: approximately 96,6 seconds for the slower approach, 465 milliseconds for second approach and 391 milliseconds with the neighborhoods' overlapping approach.

Since the slowest approach is unquestionably out of the picture, due to the computation time (it is almost 250 times slower than the overlapping approach), it has to be determined which one of the other two implemented approaches to use. Both the second approach and the one with overlapping are themselves approximations of the slowest approach and,

consequently, each of the former's steps are approximations of the latter's steps. With this in mind, relative errors of the two faster approaches ($\eta_{no\ overlap}$ and $\eta_{overlap}$) for the outputs of each step, namely the background map, the drifts and the output image were computed according to the following equations:

$$\begin{aligned}\eta_{no\ overlap} &= \frac{\|M_{slowest} - M_{no\ overlap}\|}{\|M_{slowest}\|} \\ \eta_{overlap} &= \frac{\|M_{slowest} - M_{overlap}\|}{\|M_{slowest}\|}\end{aligned}\quad (3.6)$$

where M denotes a generic matrix that is used to store the outputs of each step. The following charts (Figure 12 to Figure 14) show the results obtained using that equation in a set of 30 randomly picked fundus images.

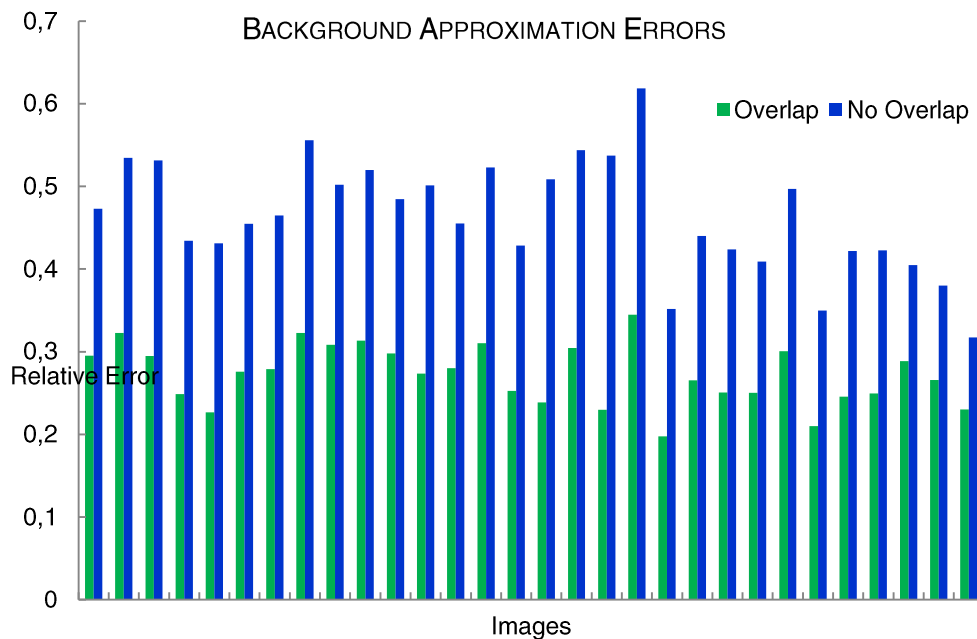


Figure 12 – Comparative chart containing the background map approximation errors for both the second approach and the one with overlap in relation to the slower approach.

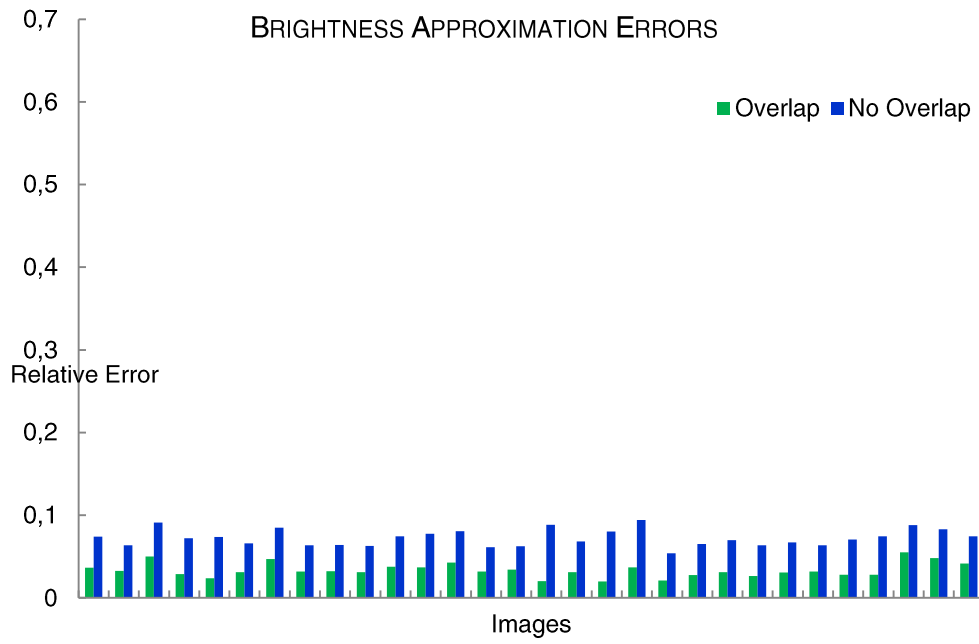


Figure 13 – Comparative chart containing the brightness drift approximation errors for both the second approach and the one with overlap in relation to the slower approach.

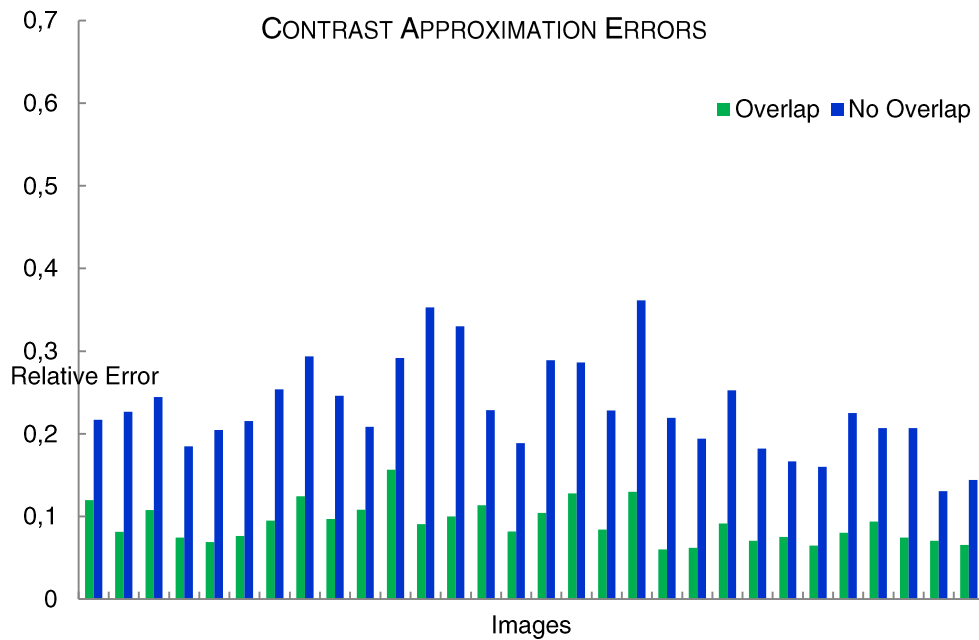


Figure 14 – Comparative chart containing the contrast drift approximation errors for both the second approach and the one with overlap in relation to the slower approach.

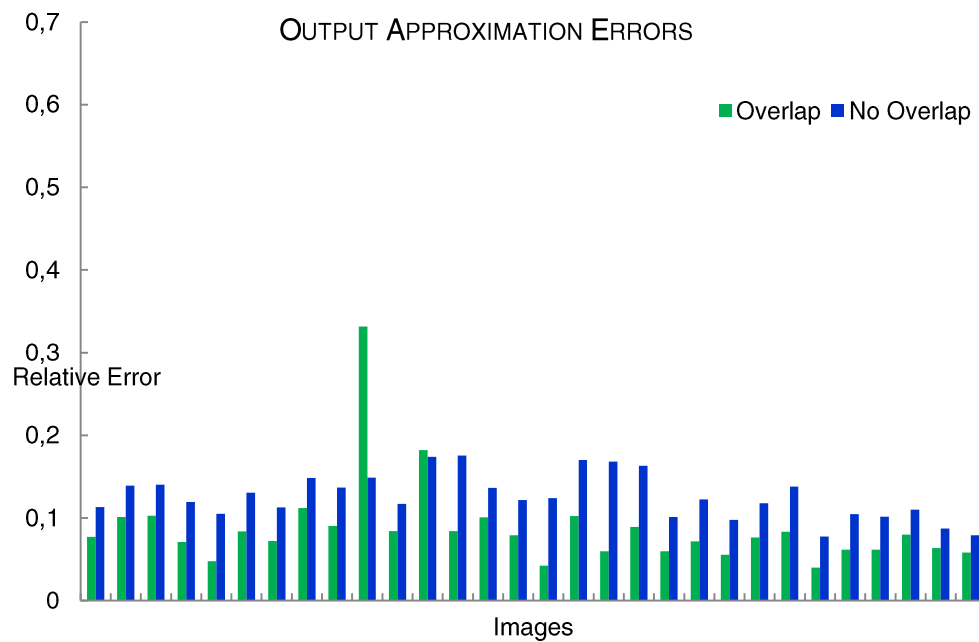


Figure 15 – Comparative chart containing the output image approximation errors for both the second approach and the one with overlap in relation to the slower approach.

Another important question that has to be clarified is which architecture to use in the processing of RGB images and if the implemented approaches are even of any use to ophthalmologists or technicians. In order to do that assessment, 10 sets of RGB images were sent to ophthalmologists, each set of images were composed of the original image and three normalized images, one for each approach (A, B or C). An example of one of these sets was already shown previously in Figure 8. The physicians were then asked to mark which approach or approaches would highlight or enhance structures and/or areas of the image that could be helpful to the diagnosis that were not totally perceptible in the original image. The responses are show in Figure 16.

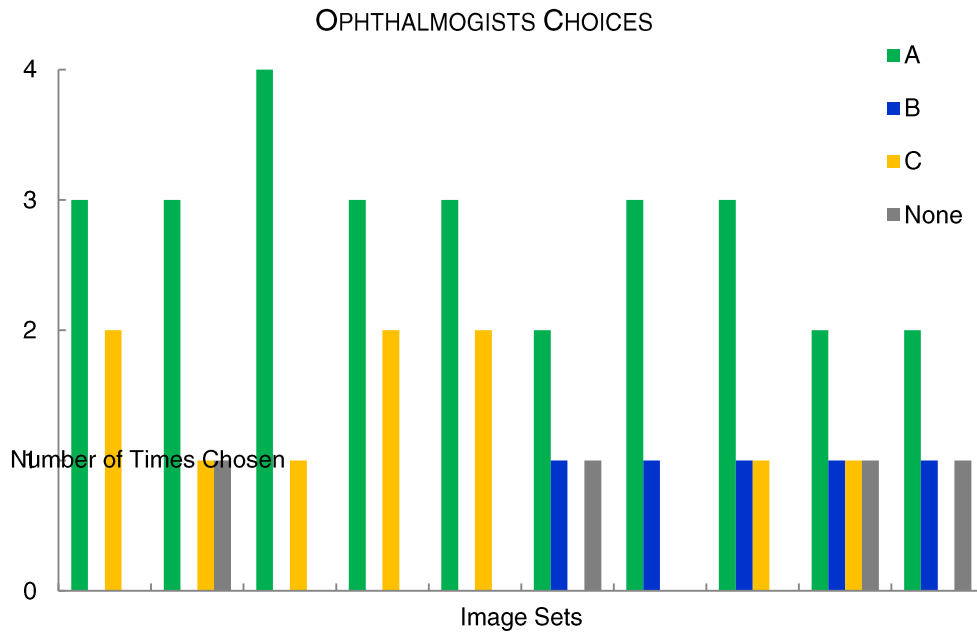


Figure 16 – Chart containing the ophthalmologists' responses.

3.2.3. DISCUSSION

Curiously, the approach with overlap turned out to be the faster than the second approach even though it has more neighborhoods in which to compute means and standard deviations. This probably happens because the interpolation process has more points to use in the computation of the remaining values.

About the approximation errors, one can easily understand that the approach with overlap is much more similar to slower approach and, except on two cases on the output image, it always succeeds to approximate better the slower approach's steps. Meaning that, in the end, the approach to be used is the one that uses overlap, since it is faster and approximates better the slower approach than the one without overlap.

Regarding the three architectures implemented for the processing of RGB images and the inquiry mentioned previously, the answers were rather diverse. However, as can be observed in Figure 16, architecture A seems to be the most useful in most cases. Nevertheless, it is not clearly that this approach is actually better than the others. In fact, from the feedback obtained from the physicians, depending on a particular area of interest, the image's acquisition conditions, the patient and the pathologies present in the image, the most useful architecture can change.

Another noticeable effect from the visual inspection of the data set used in the inquiry is that approach B seems to be very invariant to inter-patient variability. However, this could not be tested due to the limited time period of the project.

In general, the optic disk and large pathological structures are the only areas of the image that are not enhanced by this normalization technique. This happens because in those areas the neighborhoods fail to have at least 50% of background pixels, resulting in a bad estimation of the background which is propagated to rest of the steps. To address this problem, Joshi *et al.* [12] propose the use of predictable domain knowledge to manipulate the size and shape of the neighborhoods, so that, for example, in the optic disk area the neighborhoods are bigger than the macula region, maintaining, this way, an acceptable percentage of background pixels in all neighborhoods. A similar approach would be a good follow up for the implementations made in this context.

CHAPTER 4. SEMI-ASSISTED ALIGNMENT

There are several studies on fully automated registration of ocular fundus images with a wide variety of approaches and interesting result. However, those studies, due to their complexity, take years to develop and the results are not always the desired and the algorithms are, very often, computationally over expensive.

BlueWorks had already developed some previous work in this field, and two approaches were attempted: fully-automated (which did not result in a satisfactory performance) and manual. The project of creating a semi-assisted system was started in 2009, also within the scope of a Master's Thesis [13], in which a graphical user interface (GUI) was implemented to gather inputs from a technician or an ophthalmologist in order to simplify and speed up in the registration process. The system was meant to be a simpler approach to the registration problem that could be developed in a shorter time period and still could perform a robust multimodal registration. This implementation would allow the user to perform registration of two ocular fundus images, such as retinographies and angiographies, by taking a few seconds from the user that would not happen in a fully-automated system. In the other hand, a semi-assisted system would also allow a must faster algorithm to be implemented, which, in the end, could pay off for the user's lost time.

Within the scope of the same project, in order to increase the manual tool's performance, some automated features were added and the GUI was upgraded.

4.1. GRAPHICAL USER INTERFACE IMPROVEMENT

As mentioned above, a GUI was already implemented. With it, the user would load two fundus images to perform the alignment process. Next, the user had to select 6 points in the

one image and the 6 exact corresponding points in the other image. The selected points were then used to perform a second-order polynomial transformation to the second image in order to align the two images.



Figure 17 – Screenshot of the former graphical user interface.

Nevertheless, it is not an easy task to select the exact matching pixels and even an error by a few pixels would cause an erroneous registration. Consequently, considering that the goal is to consume the least user time as possible, the first step in improving this interface should address how the points are chosen. In fact, the GUI was changed and improved so that, instead of carefully selecting every point, the user would choose regions of interest (ROIs), with a variable size, containing vascular tree bifurcation areas. Then, with these regions, a processing step takes place to find a unique point for each one of them. The points that were found are then used to perform the alignment. This way, the user would not need to worry about the precision of the each entry, but only if the matching between images is correct.

Figure 18 depicts the previous and the new ways of selecting the points.

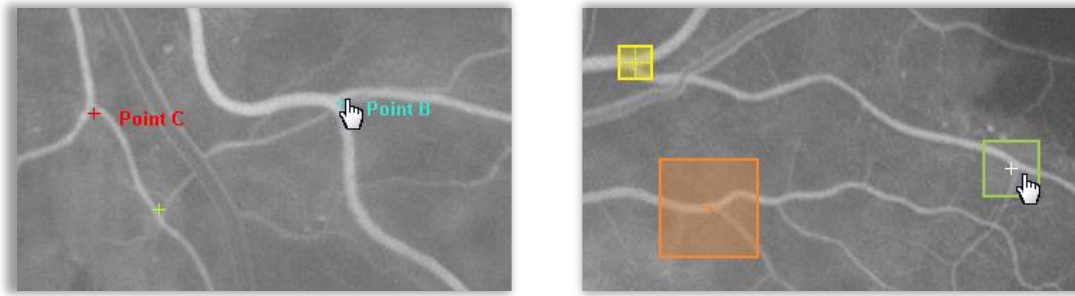


Figure 18 – Detail of both the previous and new interfaces.

Left: selecting points of interest on the previous interface. Right: selecting regions of interest in the new interface.

Moreover, in the previous interface, the area below the images was reserved to input point correspondence data. A detail of the GUI where this area is located is shown in Figure 19.

A			B		C		D		
Points	X	Y	Left		Right		Points	X	Y
Point A	310	263	Point A	↔	Point 1		Point 1	19	291
Point B	717	74	Point B	↔	Point 2		Point 2	679	51
Point C	1261	385	Point C	↔	Point 3		Point 3	979	747
Point D	1121	770	Point D	↔					
Point E	401	756	Point E	↔					
Point F	533	829	Point F	↔					

Figure 19 – Detail of the former interface showing the area where the point matching is performed. In Lopes [13].

When a point is selected in one of the images, its position information and the label assigned to it are displayed in the corresponding data table – table A (Figure 19) has information of the selected points in image to the left and table D refers to the image in the right. The matches are then established manually one by one by dragging a row of table D to table C in the correct order: first the match of Point A, then the match of Point B and so forth.

A lot of thought was put into how to upgrade the matching system in an attempt to create a cleaner and more intuitive interface. In the end, after changing the interface several times, the GUI was made to assign a color to each selected ROI, making it very easy to spot the mismatched regions and the color those should have.

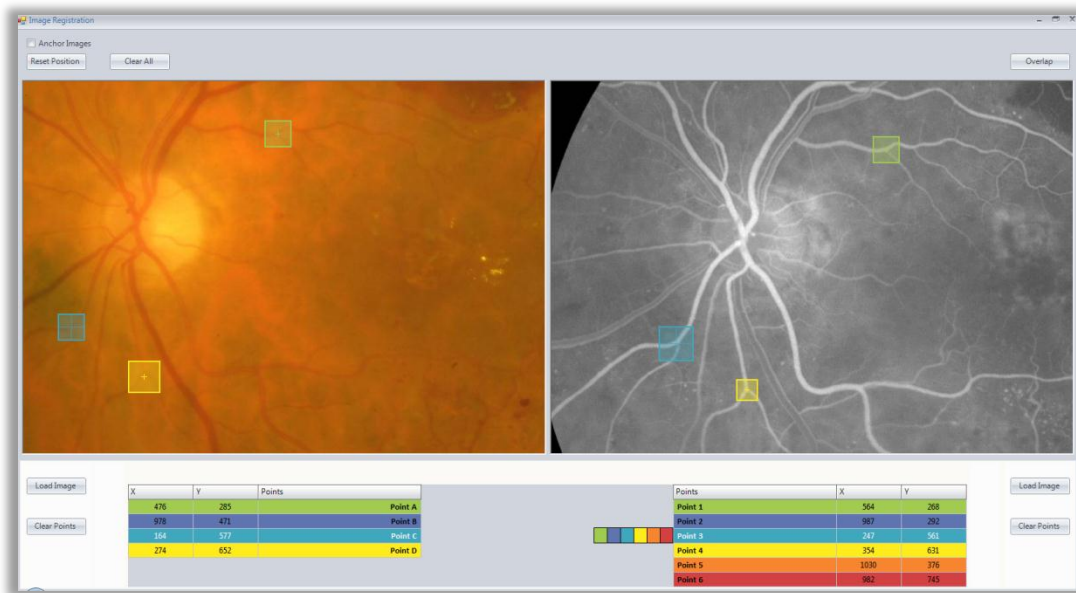


Figure 20 – Screenshot of the new graphical user interface.

Additionally, the matches between the regions are made automatically in the order that they are selected, *i.e.*, the first ROI selected in the left is assigned as the match of the first ROI selected in the right. However, a mismatch can easily be corrected by selecting the color that the mismatched ROI should have in the color palette that appears next to the right table (shown in detail in Figure 21), automatically swapping the colors between the two mismatched ROIs.

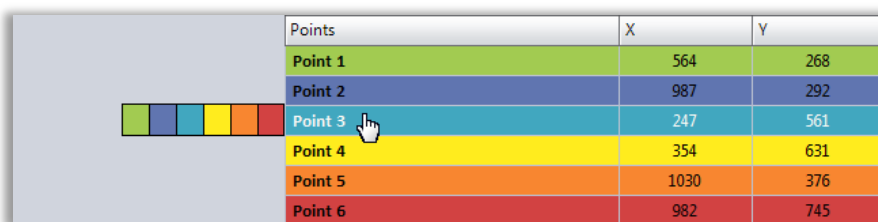


Figure 21 – Detail of the new interface showing the area where the point matching is performed.

4.2. DETECTION OF Y-FEATURES' CENTER

The next step in this semi-assisted registration system is to find the exact points to perform image registration from the defined ROIs. These points (or pixels) must be highly distinctive so that their position can be found, with a high level of certainty, repeatedly under numerous conditions that vary mainly with rotation, local distortions, illumination variability and actual divergent information between exam types. As stated before, the most static characteristics in ocular fundus images refer to the vascular tree. That is why the features chosen to be used in this processing step are vessel bifurcations or, in a more commonly used nomenclature, Y-features.

To find the bifurcation's center, it was assumed that each selected ROI correctly contains only one bifurcation and no other distinctive structures like pathologies.

Some approaches were implemented in order to find the center of Y-Features. Even so, they all started with a few simple pre-processing steps to reduce variability between ROIs. First, for RGB images, even though in the GUI still shows a 3-channel RGB image, only the green channel is used in the algorithm, since it more retinal vasculature information than any other channel (see Chapter 2). In the next step, images that have vessels darker than the background are inverted so that all ROIs have a darker background. To conclude this pre-processing stage, the ROI's histogram is scaled to fit the interval $[0,255]$, resulting in regions with similar characteristics whatever their source. For instance, in all regions, the higher level of the histogram will definitely belong to vessels, assuming that there is no other brighter structure, as mentioned above.

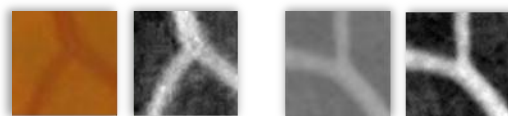


Figure 22 – Pre-processing step for the Y-features' center detection algorithm.

From left to right: retinography's original ROI and respective pre-processed ROI; ROI of an angiography with fluorescein and respective pre-processed ROI.

Vessel segmentation is a common approach to find Y-features and their centers in ocular fundus images. However, vessel segmentation, depending on the technique used, can be quite demanding. But in this case instead of an entire image, there are only a few pixels, thus the mentioned disadvantage does not have the same impact, allowing one to follow that

approach. Therefore, in the first attempts to build an algorithm to find the center of the Y-feature, vessel segmentation was used with the intention of extracting the centerlines of the vessels, which in turn would be used to easily find the center of the Y-feature.

A method called hysteresis threshold was used to perform vessel segmentation and separate the ROI into two groups of pixels (explained in detail in Appendix C). This technique uses two thresholds that are chosen to strongly characterize the two groups, *i.e.*, the higher threshold is chosen so that all pixels with intensity values above it belong to the vessels with high certainty and, regarding the lower threshold, it must be chosen to contain only pixels of the background below it. For that reason, the higher threshold was hardcoded at 240 – meaning that about 6% of the 8bit scale was assumed as vessel – and the Otsu’s threshold was used as the lower threshold (see Appendix D).

The visual assessment of several tests of the application of this method showed good results for most ROIs tested. However, there were still some ROIs being poorly segmented and, even though the bad segmentations were occurring mainly in vessels with lower caliber and in less visible vasculature, because the aim was to build a very robust algorithm, this approach was discarded and other approaches were tested. It was with a vessel tracking algorithm that the most viable results were obtained. However it was still in its first implementations when the semi-assisted alignment project was suspended. This algorithm is described in the following section.

4.2.1. VESSEL TRACKING ALGORITHM

In this algorithm, trigonometry concepts, inspired by the works of Lowell *et al.* [14], Grisan *et al.* [15] and Gregson *et al.* [16] are used to analyze consecutive profiles of each branch of the Y-feature.

The implemented approach is divided into three steps:

- Seed points extraction
- Vessel Tracking
- Center Determination

SEED POINTS EXTRACTION

During this stage, the border pixels of the ROI are analyzed and the seed points for the tracking algorithm are determined. To accomplish that, first, the border pixel with the

maximum intensity is found and it is assumed as belonging to one of the Y-feature's branches. Starting in that pixel the method searches for the left minimum and right minimum, along the borders of the ROI, according to the model illustrated in Figure 23. Afterward, the positions of the left and right edges are computed as being at half way between the intensities of the maximum and each of the respective minimum. The first seed point is set as the vessel's centerline pixel, *i.e.*, the middle point between the two edges.

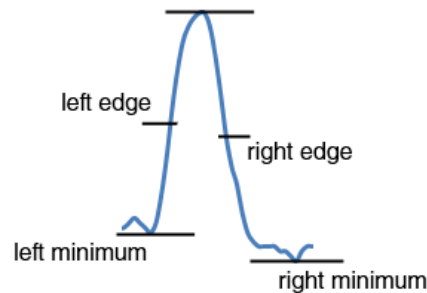


Figure 23 – Generic model of the vessel profile used in the tracking algorithm.

The second and third seed points are then consecutively searched the same way as the first.

VESSEL TRACKING

Having determined the initial seed points, the vessel tracking algorithm flows according to the following steps:

1. Find maximum value (cm_{ax_0}) in a circumference with center in the seed point (sp_0) and a radius equal to the distance between the edges (determined in the previous step when analyzing the vessel's profile);
2. Define the branch's directional vector as $v = cm_{ax_0} - sp_0$;
3. Find new centerline pixel, using the profile model, along the line segment bisector between sp_0 and cm_{ax_0} ;
4. Define more seed points along a line segment from sp_0 to the new centerline pixel;
5. Store new centerline pixel as a seed point;
6. Go to the next seed point (sp_n);
7. Find new maximum value (cm_{ax_n}) in a circumference with center in the seed point (sp_0) with a slightly larger radius (search conditioned by v);
8. Define the branch's directional vector as $v = cm_{ax_n} - sp_n$;

9. Find new centerline pixel, using the profile model, along the line segment bisector between sp_n and $cmax_n$;
10. Store new centerline pixel as a seed point;
11. Return to step 6.

The vessel tracking loop continues for the three branches simultaneously, until, through a series of conditions, the three parallel tracking converge in the center of the Y-feature.

CENTER DETERMINATION

The final stage consists in doing a linear regression of the last centerline pixels extracted for each branch and finding the intersections between the three lines. The center of the Y-feature is then determined to be the pixel in the exact middle of the tree intersections.

4.3. DISCUSSION

This line of work was suspended without the possibility of doing other intended modifications and improvements. If this architecture were to be followed, the next steps would certainly be to use more robust seed points and a different model. A pre-processing step with Gaussian filters would undoubtedly produce more reliable seed points and would also reduce vessel tracking errors. Regarding the profile model used, there are a few others that would be worth being implemented such as the model HHFW (half-height at full width), also known as full width between two half minima (FWHW), which is a reliable method for vessel diameter assessment [17] that also accounts for light reflex in the retinal arteries. Other interesting model that appears to have very good results is the model published by Lowell *et al.* [14], which consists in a two-dimensional model composed by the difference between two Gaussian curves that is fitted to the vessel's profile.

The brightness and contrast normalization algorithm discussed in the previous chapter did not seem to have much impact in the determination of the Y-features' center. Which is partly understandable, since these ROIs are too small to have great discrepancies in contrast and brightness.

CHAPTER 5. FULLY-AUTOMATED MULTIMODAL REGISTRATION

This chapter aims to create a fully-automated system to perform registration of ocular fundus images from multiple imaging systems and acquisition methods based on well-known algorithms, such as the SIFT (Scale Invariant Feature Transform) [2] and SURF (Speeded-Up Robust Features) [3] methods.

The overall architecture of the algorithms that were implemented in this part of the project is represented in Figure 24.

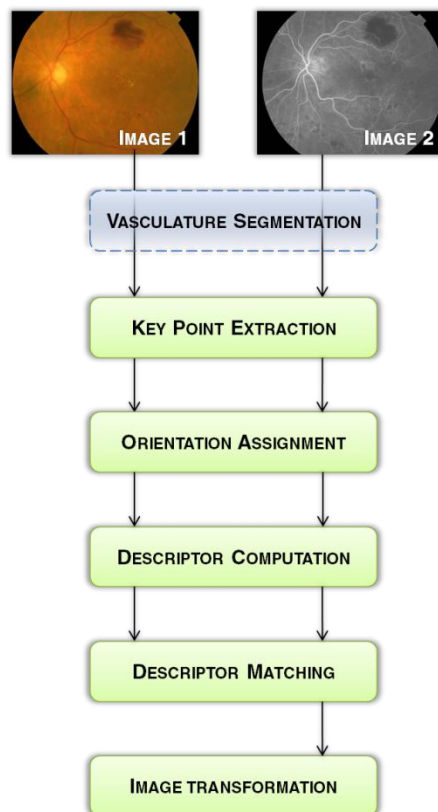


Figure 24 – Representative flowchart of the registration algorithms implemented.

Two distinct architectures were implemented, the main differences being the type of descriptor used in the descriptor computation step and the fact that the first architecture does not involve performing vessel extraction. In addition, in both solutions, only the green channel is considered when processing RGB images, since it has more contrast and contains more useful information (see chapter 2).

Each module of the algorithms was implemented in Visual C++ and is going to be discussed throughout this chapter. Furthermore, an evaluation of the overall algorithms' performance is also carried out.

5.1. INTEGRAL IMAGE

In order to greatly decrease computation effort regarding the search of points of interest as well as descriptor computation, there is the need to build an integral version of the source image. This image is computed using the following equation:

$$I_{\Sigma}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \quad (5.1)$$

resulting in a matrix where each entry represents the sum of all pixels of the input image (I) before that entry, *i.e.*, each entry is the sum of the pixels in a rectangular region defined by two opposite vertices, one in the first pixel of the image and other in that entry's position. That matrix is called integral image (I_{Σ}).

After having computed the integral image, to calculate the sum of intensity values in any rectangular region (Σ_R), in an upright position (see Figure 25), it only takes three sum operations whichever its size, via the following equation:

$$\Sigma_R = I_{\Sigma}(x_2, y_2) - I_{\Sigma}(x_2, y_1) - (I_{\Sigma}(x_1, y_2) - I_{\Sigma}(x_1, y_1)) \quad (5.2)$$

in which x_1 , x_2 , y_1 and y_2 refer to the coordinates of the vertexes as demonstrated by Figure 25.

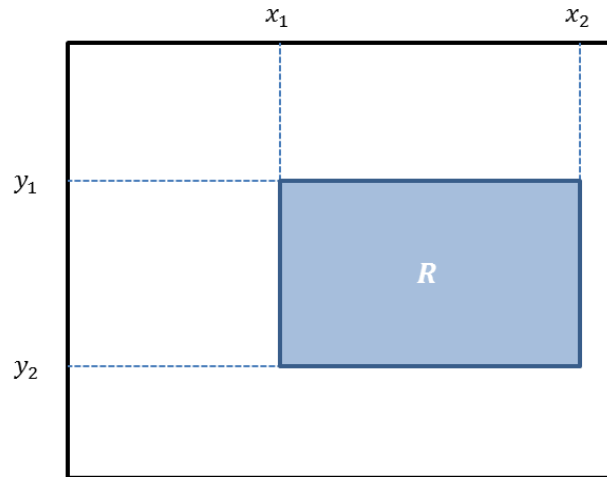


Figure 25 – Diagram of a generic rectangular region in an upright position.

This approach is particularly useful for scale-space analysis, which will be further discussed in the next section.

5.2. POINTS OF INTEREST EXTRACTION

To find points of interest in a given image the approach chosen was the one described by Bay *et al.* in their SURF paper [3]. This approach uses the determinant of the Hessian matrix at multiple scales to find key features throughout the image.

5.2.1. HESSIAN MATRIX DETERMINANT LAYERS

The first step in this searching mechanism is to build a set of matrices or images where each matrix represents data at a different scale and each entry is the Hessian's determinant of that particular point.

The Hessian contains second-order partial derivatives of a given function. However, in this case this function is replaced by the discrete three-dimensional function that describes the pixel intensities throughout the image. Therefore, there is the need to adapt the way the second-order derivatives are computed.

As mentioned by several authors, the convolution of the input image with Gaussian second-order derivatives seems to be optimal for scale-space analysis [3] and is used with

good efficiency in the SIFT algorithm [2], but their lack of performance encourages other approaches [18]. The SURF searching mechanism approximates the Gaussian kernels to box filters allowing the algorithm to take advantage of integral image properties.

Figure 26 compares the Gaussian and the adapted kernels.

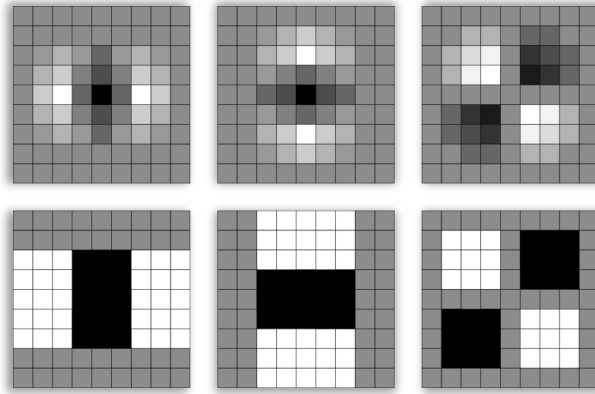


Figure 26 – Templates for computing the image's intensity second-order derivatives. In Evans [18].

Top: Gaussian templates. Bottom: weighted box filters. From left to right: templates for xx -directions, yy -directions and xy -directions.

The method used to weight the new box templates is quite simple. The weights are assigned in order to obtain a balanced distribution. In fact, their assignment is done to assure that their DC response is null. In other words, they will be sensitive only to intensity variations and not in areas without texture, as intended. With this in mind, on the first two templates of the bottom row the white boxes are weighted +1 and the black -2 and on the last one the white boxes are weighted +1 and the black ones -1 . All gray areas are weighted zero and do not even become part of the equations.

As one should already have realized the convolution process can now be reduced to a few sums per iteration with the use of equation 5.2.

The pyramidal process that is frequently used [18] to create the determinant layers is also modified. Usually, the image is sequentially smoothed and sub-sampled to create a pyramidal set of images that are then convolved with the Gaussian second-order derivative templates resulting in the needed information to compute the Hessian. However, instead of creating a set of images with different sizes by down-scaling the original image and then use the same kernels, since the computation effort of convolving with the adapted kernels is independent from their size, the algorithm is made to create a set of templates with different sizes, each of

them corresponding to a different scale, and to compute the Hessian by convolving them with the same image in its original size (see Figure 27).

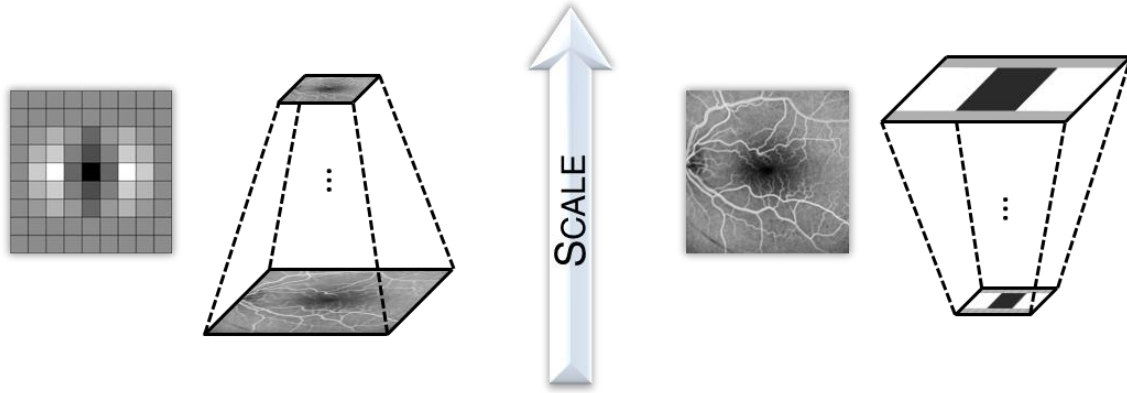


Figure 27 – Scheme showing a typical pyramidal process (left) and the used pyramidal process (right).

Adapted from Bay *et al.* [3].

The Hessian matrix for a scale s and a point $p = (x, y)$ now becomes:

$$H(p, s) = \begin{bmatrix} D_{xx}(p, s) & D_{xy}(p, s) \\ D_{xy}(p, s) & D_{yy}(p, s) \end{bmatrix} \quad (5.3)$$

where D_{xx} , D_{xy} and D_{yy} are the image's second-order derivative approximates that result from the convolution process of with new templates. D_{xx} , D_{xy} and D_{yy} are, therefore, also the Laplacian of Gaussian (L_{xx} , L_{xy} and L_{yy}) estimates (see Appendix B).

After their computation, D_{xx} , D_{xy} and D_{yy} responses are normalized according to the size of the filter's used.

With equation 5.3, computing the Hessian's determinant (det_H) is very straightforward:

$$det_H(p, s) = D_{xx}(p, s)D_{yy}(p, s) - (\omega D_{xy}(p, s))^2 \quad (5.4)$$

with $\omega = 0,9$ [3], which is a correction factor needed to balance the equation due to the approximations in the kernel. This expression is used to build a set of 10 layers, one for each scale. Moreover, 3 pyramidal sets of templates are needed – one set for the second-order partial derivative in xx -directions, one for the yy -directions and the last for xy -directions.

The size increasing mechanism separates the templates in 4 octaves, each octave with 4 different size values. The first octave starts with a 9x9 kernel and has step of 6 pixels, resulting

the following sizes: 9x9, 15x15, 21x21 and 27x27. The second octave starts with the second kernel of the previous octave and the step doubles its value. This repeats for the remaining three octaves. For better understanding of sizes of the filters used and their distribution across the octaves see Figure 28.

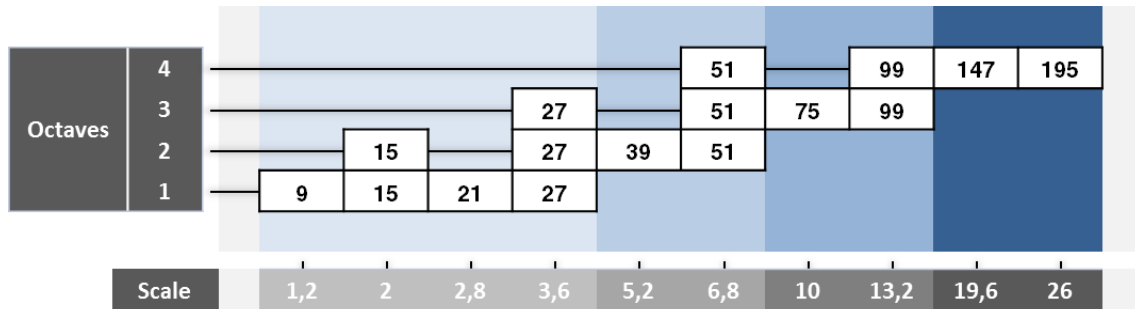


Figure 28 – Representative scheme of the distribution of the kernel's size across the octaves and their relation with the scale they are intended to.

For simplicity, from now on, the layers are going to be referred to by the size of the template used for its creation with the prefix T-. For instance, the first layer is going to be called layer T9.

5.2.2. ACCURATE MAXIMA LOCALIZATION

Having computed all response layers of the pyramid, the algorithm starts the actual search for the key points. First, the response values are cropped using a pre-determined threshold. Then, the layers are searched for local maxima. A point is determined as a point of interest if its response value is greater than its 26 neighbors, 8 in its layer and 9 in the adjacent layers above and below as illustrated in Figure 29.

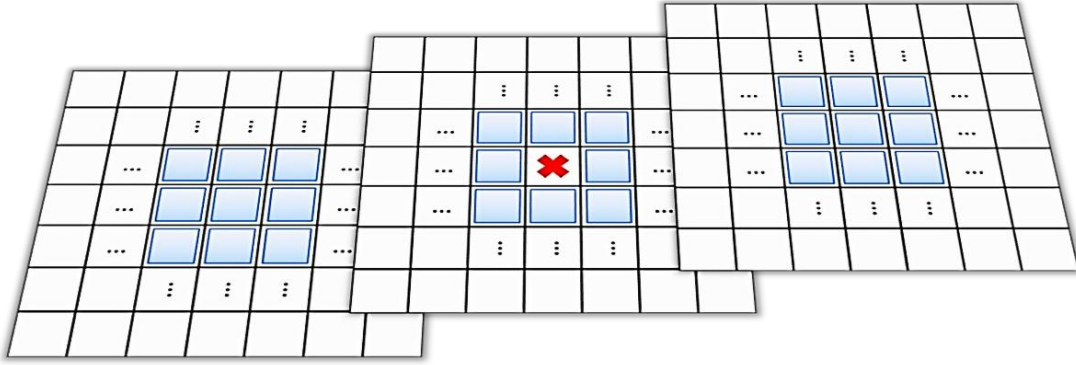


Figure 29 – Illustration of the search for local maxima in a generic point (red mark) by comparing it to its 26 immediate neighbors (blue squares). Adapted from [2].

Note that for each octave there is only 2 local maxima searches. For instance, for the first octave there is a search for local maxima throughout layer T15, using also layers T9 and T21, and throughout layer T21, using also information from layers T15 and T27.

In the final stage of the process of extracting key points, the data around it is fitted to a 3D quadratic curve, in order to find the location of the point with sub-pixel accuracy, in both scale and space. This is particularly important to find the actual scale, since scale sampling becomes very large in the higher octaves. [3, 18]

Considering a Taylor expansion up to quadratic terms centered at the detected discrete location of the key point $p = (x, y, s)$:

$$\det_H(p) = \det_H + \frac{\partial \det_H}{\partial p} p + \frac{1}{2} p^T \frac{\partial^2 \det_H}{\partial p^2} p \quad (5.5)$$

One can find the new location at the extremum of the quadratic curve, by computing:

$$\hat{p} = - \frac{\partial^2 \det_H^{-1}}{\partial p^2} \frac{\partial \det_H}{\partial p} \quad (5.6)$$

If \hat{p} is greater than 0,5 in any direction the point will be discarded, so that only the most stable points are marked as points of interest.

The threshold used to crop the response values is adapted and the search process is repeated until a desirable number of points are found.

5.3. ORIENTATION ASSIGNMENT

With the purpose of obtaining orientation-invariant descriptors, these will have to be computed according to a local orientation. The method implemented in this context to obtain a reproducible local orientation using the integral image properties is described by Bay *et al.* [3].

The method starts by defining a circular region with a radius of $6s$ centered in the key point in study, where s refers to the scale of the key point, detected in the previous section (see Figure 30).

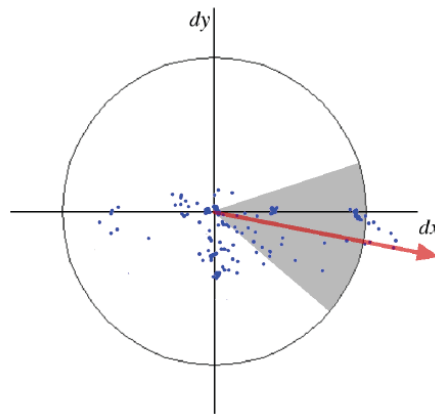


Figure 30 – Orientation assignment illustration showing the response vectors (blue points), the angle aperture being analysed (darker region) and the assigned orientation (red vector). In Bay *et al.* [3].

Afterward, the region is sub-sampled using a step size of s and the response of the Haar wavelet (shown in Figure 31), with a side length of $4s$, is computed for those sampled points, creating a response vector ($r_{Haar} = (r_x, r_y)$) for each point. Like before, the integral image can be used to speed up the process of box filtering.

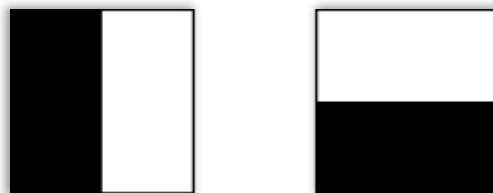


Figure 31 – Haar wavelet filters with black and white regions weighted with -1 and +1, respectively. Left: filter to generate response in x direction. Right: filter to generate response in y direction.

All response vectors are weighted based on the relative position of the equivalent sample point according to a Gaussian curve with a standard deviation of $2,5s$. The response vectors are then searched for a dominant orientation. To do that, the sum of all vectors, within an aperture of $\frac{\pi}{3}$ is computed, represented as the darker region of Figure 30. Then, the aperture is slightly rotated and sum is computed again. After all vector sums are computed, the biggest vector sum is chosen as the dominant orientation.

5.4. SIFT DESCRIPTORS AND MODIFICATIONS

At this point each key point has a set of 2D coordinates, a scale value and an orientation assigned to it. The next task in the algorithm's course will address the creation of a descriptor for each of the points of interest. This descriptor must be robust enough that it could be accurately reproduced in another image with that key point in common. However, this can be a delicate path, since the intention of the whole registration algorithm is to align images from a wide variety of imaging systems and methods of image acquisition.

As discussed before, a pair of images from two different exams contains countless differences, caused primarily by contrast and brightness variations, as well as variations provoked by the angle of acquisition and other local distortions like geometrical distortions triggered by the saccadic eye movements during image acquisition. Additionally, in some pairs of images, other problems arise, like the inversion of contrast and, even more problematic, different visible information caused by the different nature of imaging techniques, *e.g.*, between a retinography and an angiography. This difference in the visible information of different methods of acquisition is what motivates the creation of a multimodal registration system, but it is also the greatest obstacle.

In the next subsection of this thesis (5.4.1) the descriptors proposed by Lowe [2] are going to be described followed by the explanation of other similar approaches that were implemented in this project (5.4.2 and 5.4.3).

5.4.1. SIFT DESCRIPTORS REVIEW

SIFT descriptors are generated by computing the gradients around the key point's location weighted by a Gaussian curve with $\sigma = 8$. However, one must consider the previously computed dominant orientation of the key point if the intention is to create orientation invariant descriptors. Therefore, the gradients are computed in a quadrangular area (16x16 pixels) centered in the point of interest, but, instead of an upright position, the square will be rotated by an angle equal to the orientation assigned to the key point. Moreover, to obtain scale invariance the gradients will have to be computed taking into account the scale of the point. Lowe [2] uses Gaussian blurring to impose scale information in the descriptor. After having computed all gradients within the square along the x and y directions, they need to be recomputed to match their new axis in order to achieve orientation invariance.

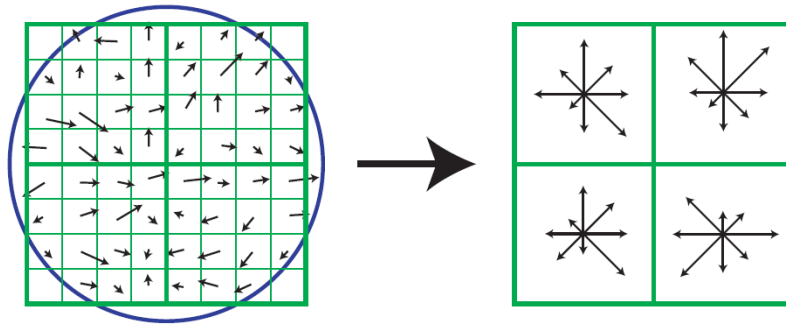


Figure 32 – Rotated gradient samples in an 8x8 quadrangular region around a key point and their distribution in the corresponding 2x2 histograms. In Lowe [2].

The blue circumference represents the Gaussian ($\sigma = 4$) used to weight the gradients. This is a representative scheme. In fact, the algorithm uses a Gaussian with $\sigma = 8$, 16x16 samples and 4x4 histograms.

The next stage consists in clustering the samples in the square in 16 (4x4) smaller squares and using gradient information in each cluster to build an orientation histogram. Each histogram has 8 bins with a $\frac{\pi}{4}$ angle range each (for example $[0, \frac{\pi}{4}[$, $[\frac{\pi}{4}, \frac{\pi}{2}[$, ..., $[\frac{7\pi}{4}, 2\pi[$).

The location of each bin's center, as well as of a gradient sample, can be described as a set of 3D coordinates. The first two coordinates refer to the position in the larger square. The last coordinate refers to the angles. The histograms are generated by summing the value of each gradient's magnitude to the respective bin, weighted according to the distance of the sample to center of the bin. However, the value is also spread to the adjacent bins (up to 3, one for

each coordinate), also weighted according to distance to those bins. These weights are computed using linear interpolation.

The result is a set of 16 vectors with 8 elements each that can be rearranged into a single vector of 128 elements. This vector is then normalized to unit length to make it invariant to contrast changes. Brightness variations do not cause any problem since the descriptor is computed from pixel differences.

The previous step only addresses linear illumination changes. However, there are also some variations caused by non-linear illumination problems that can greatly change some gradient magnitudes [2]. Therefore, to attenuate this effect, the larger gradients magnitudes are thresholded to have no value greater than 0,2. This reduces the influence of gradient magnitudes and gives emphasis to the distribution of orientations.

To conclude the computation of the descriptor, the vector is renormalized to unit length.

5.4.2. ARCHITECTURE I – MODIFIED SIFT DESCRIPTORS

The descriptors proposed in this section are an adaptation of both descriptors proposed by Lowe [2] and Chen *et al.* [19].

The first major modification to SIFT descriptors is the way of computing the gradients of the samples. As mentioned before, SIFT descriptors compute their gradients simply convolving differentiation templates with a blurred image. The blur is applied using Gaussian templates with a standard deviation proportional to the detected scale of the point of interest. Chen *et al.* use a similar way of computing the gradients but the descriptors they propose are not scale-invariant; thus, scale assignment is not even performed. The modified descriptors here suggested use gradients computed with the same method used before (in section 5.3) where Haar wavelet responses along x and y replace the gradients, and the scale is used to choose the size of the templates. This way, one is able to take advantage of the processing speed of box filtering based on the integral image.

The method used to compute the histograms is the same as the one used by Chen *et al.*, [19], which is an adaptation of the SIFT descriptors that is designed to be invariant to inversion of contrast, such as in what happens to the vascular trees between retinographies and angiographies. The result is what Chen *et al.* call a symmetric descriptor. To compute this descriptor, first the gradients are forced to be comprised in $[0, \pi[$ by rotating all sample gradients within $[\pi, 2\pi[$ by an angle of $-\pi$ (see Figure 33). The histograms still have 8 bins, consequently, the angle range of each bin has to be reduced by half, *i.e.*, the angle range is

reduced to $\frac{\pi}{8}$. This generates a $4 \times 4 \times 8$ sub-descriptor (A). Then, after normalizing sub-descriptor A to unit length, another sub-descriptor (B) is computed from the first one using the following equation:

$$B(i, j, k) = A(5 - i, 5 - j, k) \quad (5.7)$$

where $i, j = 1, 2, 3, 4$ refer to the histogram's position and $k = 1, 2, \dots, 8$ refers to the bin's position within each histogram.

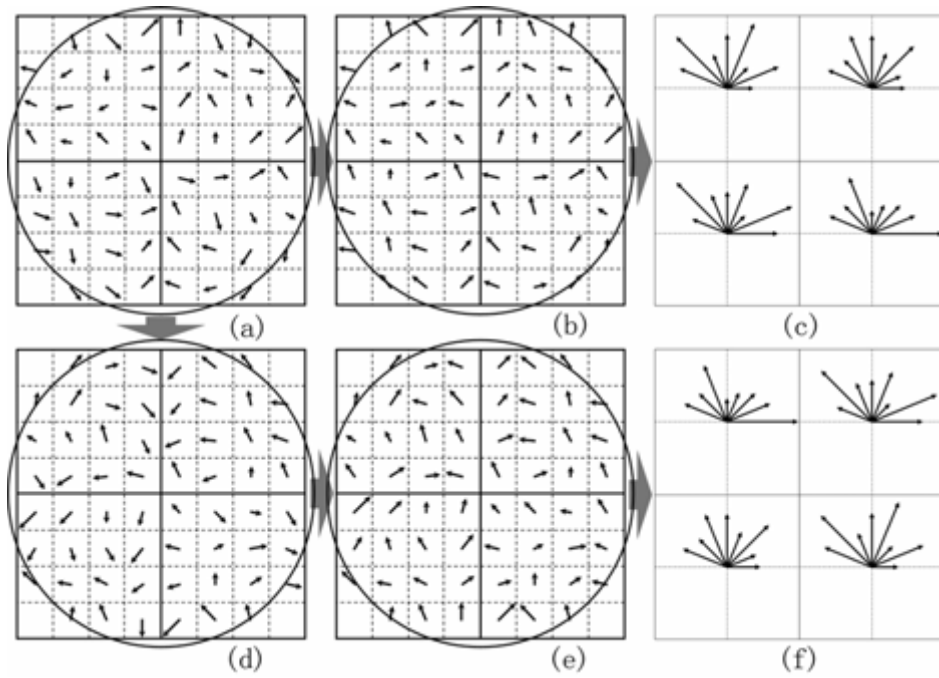


Figure 33 – Illustration of the computation of a symmetric descriptor. In Chen *et al.* [19].

Key: (a) sampled gradients; (b) gradients induced to be comprised in $[0, \pi[$; (c) histogram representation of sub-descriptor A ; (d) to (f) demonstrates the computation of sub-descriptor B . Note that this is a representative diagram and the number of samples and bins do not correspond to the reality.

Having both sub-descriptors, the final descriptor (dsc) can be computed according to:

$$dsc(i, j, k) = \begin{cases} c_1 |A(i, j, k) + B(i, j, k)|, & i = 1, 2 \\ c_2 |A(i, j, k) - B(i, j, k)|, & i = 3, 4 \end{cases} \quad (5.8)$$

where c_1 and c_2 are adjustable parameters that were set to $c_1 = 0,5$ and $c_2 = 1$, simply to make each element of dsc have the same range.

5.4.3. ARCHITECTURE II – SEGMENTATION AND SIFT

Architecture I was the first of the two to be implemented and its goal was to combine scale-space analysis with the symmetric descriptors proposed by Chen *et al.* [19]. Nevertheless, the first tests of that architecture showed that it did not perform well on image pairs with greater heterogeneity. Therefore, the idea to use the well-known SIFT descriptors with approximations for speed improvement and a previous step of vasculature segmentation arose.

Like in the modified SIFT, here, it is used the Haar wavelet responses to replace the gradients and the scale is used to choose the size of the templates, to take advantage of box filtering using the integral image.

Several approaches to perform multimodal registration with ocular fundus images use vessel segmentation as an early step. Nevertheless, the segmentation process is not a trivial task and there are several studies that try to overcome this problem [20-22]. Sometimes they are successful, but usually their algorithms succeed with only a few types of images or at least they do not report their robustness across a wide variety of exam types.

In this project, the first approach to perform vessel segmentation involved transposing to Visual C++ an existent solution developed by *BlueWorks* [13], which was itself an adaptation of an algorithm by Chanwimaluang *et al.* [11, 23]. Summarizing, this implementation uses matched Gaussian filters to pre-process the image in study and then uses a global entropy-based threshold to differentiate vessel from non-vessel pixels. The pre-processed image results of the convolution of the image with a set of 12 rotated Gaussian templates and taking the maximum value for each pixel along the 12 convoluted images. The use of Gaussian templates is predicated on the assumption that all vessels have a profile similar to a Gaussian curve.

There are two types of templates that can be used in the segmentation process and they are chosen according to the exam type in study. If the image has vessels brighter than the background, as it does on angiographies, then the templates used result from the rotation of the template depicted on the right of Figure 34. Although, if the image contains a brighter background, much like what happens in retinographies, the template set used is based on the template on the left. The set of 12 templates is generated by sequentially rotating the main template by an angle of $\frac{\pi}{12}$.

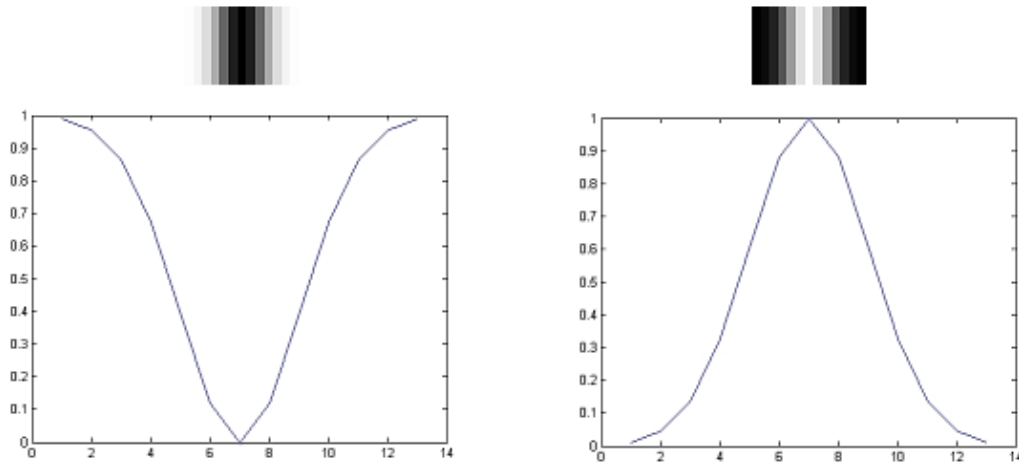


Figure 34 – The two template types that are used to pre-process ocular fundus images. Adapted from Lopes [13].

Nevertheless, this approach was not robust enough to perform vessel segmentation to the fundus images to which it was intended. In fact, it turned out to be very instable and even small lesions would affect its performance.

In another attempt to perform segmentation, an algorithm already used by Martinez-Perez *et al.* [24] was implemented, this time, however, it was simply tested in MATLAB® language. In this technique the gradients and the Hessian matrix (Appendix B) are computed along several scales by changing the value of σ in the Gaussian kernels. Afterwards, from the eigenvalues of the Hessian matrix, one can compute the principal curvature values, resulting in a set of matrices or representative images with gradient magnitudes and principal curvature values. With these two resultant matrices a set of consecutive stages is then performed where a process called hysteresis threshold is applied. For more information on the hysteresis threshold see Appendix C.

However, the results obtained with this approach were not also the desired. This implementation succeeds with great performance in some images, but fails to classify the vasculature on other images.

5.5. DESCRIPTOR MATCHING

The whole point of having descriptors of an image's points of interest is to have them match the descriptors from another image in order to obtain point-to-point correspondences between a pair of images which can then be used to perform an alignment transformation.

The algorithm for descriptor matching starts by proceeding to the comparison between the descriptors from the two images that are going to be registered. The metric used for comparison was the Euclidean distance:

$$d_{Euclidean}(i, j) = \sqrt{(dsc_i^1(1) - dsc_j^2(1))^2 + \dots + (dsc_i^1(128) - dsc_j^2(128))^2} \quad (5.9)$$

with $i = 1, \dots, n_1$ and $j = 1, \dots, n_2$, where n_1 and n_2 are the number of points of interest in the first and second image, correspondingly. dsc_i^1 is the descriptor of the key point i of the first image and dsc_j^2 the descriptor of key point j of the second image. The group that contains all descriptor from the first image is going to be called \mathcal{D}_1 and for the second image the terminology will be \mathcal{D}_2 .

First, for each descriptor of \mathcal{D}_1 , the algorithm is going to find the most probable candidate for a match in \mathcal{D}_2 , i. e., the descriptor from \mathcal{D}_2 with the lowest Euclidean distance. However, only the strongest matches continue to the next phase of the matching algorithm. In fact, the matches are cropped by what is usually known as ambiguity threshold [25]. This threshold is used to eliminate the matches that have a low ratio between the distance of the closest match and the distance of the second closest.

$$d_{ambiguity}(i) = \frac{d_{Euclidean}(i, d_1)}{d_{Euclidean}(i, d_2)} < 0.65 \quad (5.10)$$

with d_1 being the index of the closest match and d_2 the index of the second closest.

In the next step, the same is done but with the descriptors from \mathcal{D}_2 , resulting in two groups of matches, one that refers to the strongest matches of \mathcal{D}_1 in \mathcal{D}_2 and another that contains the strongest matches of \mathcal{D}_2 in \mathcal{D}_1 . These two groups are then merged together prevailing only the matches that exist in both groups. This is called bilateral matching [19].

Only the strongest matches are considered at this point, nevertheless, there are usually still some wrong matches, especially with challenging pairs of images. Therefore, the goal of the

next step is to eliminate all wrong matches. The elimination of false positive matches is done in two steps, according to descriptors' orientation and according to key points' relative position. The first is based on the fact that the differences between the orientations of each match are approximately constant [19]. Thus, if a match's orientation difference differs greatly from the median of all orientation differences, that match is discarded.

As regards to false positive elimination according to relative positions, it is predicated on the assumption that there is no great affine transformation in the pair of images [19]. Knowing that at this point there is one list of n elements, each element containing a match (composed by m_1 , from the first image, and m_2 from the second image), the algorithm follows the following steps:

1. Compute every match distance ratio according to:

$$r_{pos}(i, j) = \frac{d_{Euclidean}(m_1(i), m_1(j))}{d_{Euclidean}(m_2(i), m_2(j))}, i, j = 1, \dots, n \quad (5.11)$$

2. Calculate the median of all n^2 values of r_{pos} ;
3. For each value of i , compute the n mean values of the differences between r_{pos} values (along j) and the median;
4. Eliminate matches that have a high mean.

In other words, it assumes that the ratio between Euclidean distances of the feature points of on one image and the distances of the matched features on the other image is approximately constant. If there is a match that has ratios too different from the median value, that match is excluded.

5.6. TRANSFORMATION

Regarding the transformation process of the second image, to perform the alignment of both images, a second-order polynomial transformation process is used (equation 5.12), since it had been already implemented in a *BlueWorks* project [13].

$$X_C = \begin{bmatrix} a_{00} & a_{10} & a_{01} & a_{11} & a_{20} & a_{02} \\ b_{00} & b_{10} & b_{01} & b_{11} & b_{20} & b_{02} \end{bmatrix} \begin{bmatrix} 1 \\ x_D \\ y_D \\ x_D y_D \\ x_D^2 \\ y_D^2 \end{bmatrix} \quad (5.12)$$

where X_C refers to the correct position of the points, or this case, the positions in the first image, and $X_D = (x_D, y_D)$ is position of the points in second image. a and b are coefficients that have to be pre-computed with the extracted matches from the previous stage.

Note that to compute a and b coefficients, one only needs 6 matched feature points. So in order to select 6 matches from the set of matches retrieved from the previous stage, a K-means algorithm was used to gather the matches' points relative to the first image into 6 clusters. The points closer to the center of each cluster and their matches were then extracted to compute the transformation matrix.

5.7. RESULTS

Performance evaluation of multimodal registration algorithms is a difficult process since there are few algorithms available for comparison that perform such task. However, one algorithm which is commonly used for comparison with this type of implementations is the Dual-Bootstrap Iterative Closest Point algorithm [26]. This algorithm will also be used here for comparison with the implementation of the first architecture (symmetrical descriptors).

A set of 12 randomly chosen multimodal pairs of images is used to evaluate the performance of the first architecture. The results are shown on Figure 35.

Note that the results obtained using the approaches described in this thesis are simply the matches that result from the bilateral matching and not the results from a final alignment. This compromise solution, caused by the limited time frame of the project, enables one to assess the performance without the influence of the false positive elimination process.

Furthermore, the results for the first architecture were obtained using a higher value for the ambiguity threshold since this method seems to produce less distinguishable descriptor due to being symmetrical. Therefore, the matches are less robust, producing a more unstable registration process.

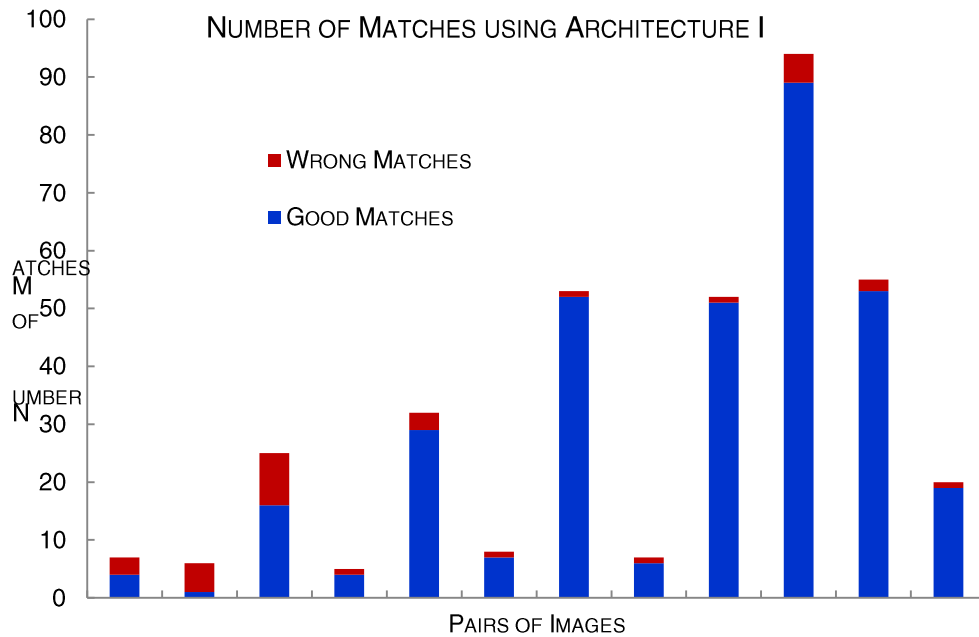


Figure 35 – Chart containing the number of matches using architecture I (symmetric descriptors)

Regarding the dual-bootstrap algorithm, it was able to perform registration to 6 of the 12 pairs. Therefore, aiming for a second-order polynomial transformation (which only needs 6 matches to be computed), theoretically, this architecture would perform better, since it has more than 6 correct matches in 8 out of the 12 pairs. However, this is not a fair comparison because one still needs to perform false matches elimination.

Regarding architecture II, since there were no segmentation algorithm implemented that was robust enough to validate these descriptors, a way of performing vasculature segmentation external to *BlueWorks* was needed. The most precise way of evaluate the performance of these descriptors were to have a set of image pairs segmented, manually or with another algorithm sufficiently robust to segment the wide range of exam types for which they were intended to. Nevertheless, none of the mentioned approaches was possible to execute. Therefore, another path was taken.

The DRIVE (Digital Retinal Images for Vessel Extraction) database [27] contains a set of fundus images and their manually created segmentation maps by two trained human observers. However, the dataset does not contain pairs of images so, in order to simulate two different images, two other datasets were created from one of the manual segmentation sets according to the following:

1. A dataset was generated by performing a random perspective transformation to the segmented images from the second observer;
2. A second dataset was generated by arbitrarily erasing vessels from the first dataset, simulating poorly segmented images (see Figure 36).

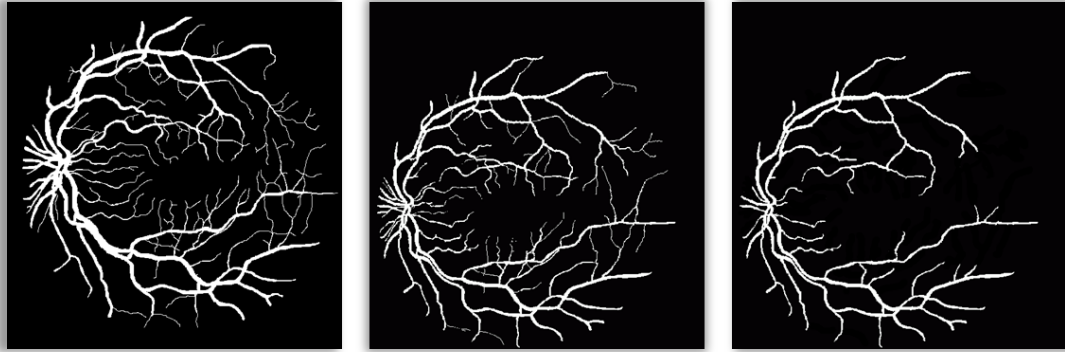


Figure 36 – Example of the images created from the DRIVE database.

From left to right: original map created by the first observer; map from the second observer with perspective transformation; map with perspective transformation and vessel deletion.

Summarizing, now there is a dataset from the first observer and other two generated from the second observer which are going to be registered with the first for evaluation of the architecture II. The results are shown in Figure 37 and Figure 38.

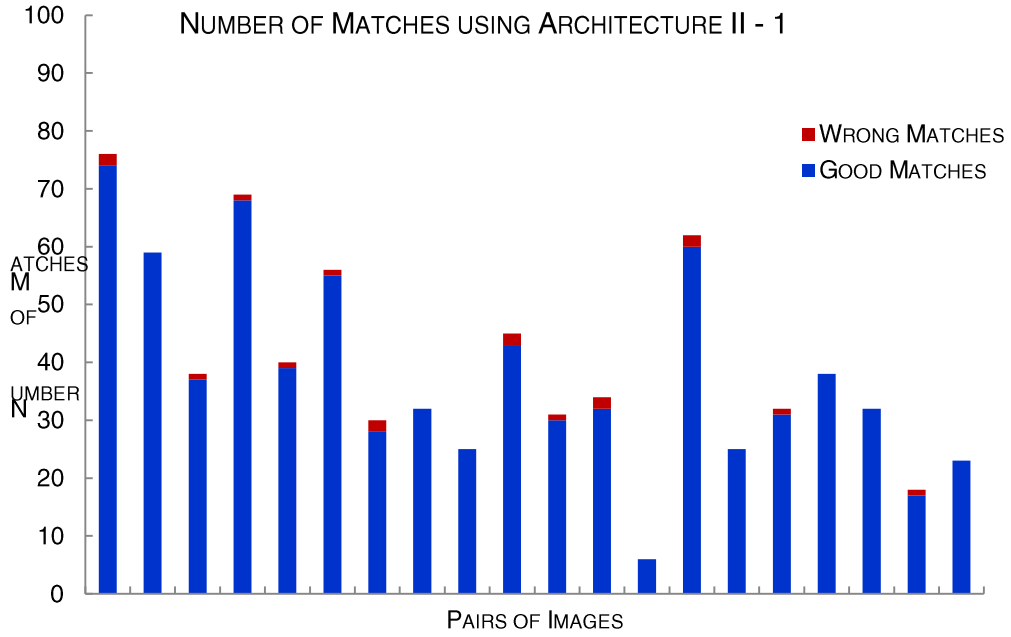


Figure 37 – Chart containing the number of matches using architecture II on the data set with perspective transformation.

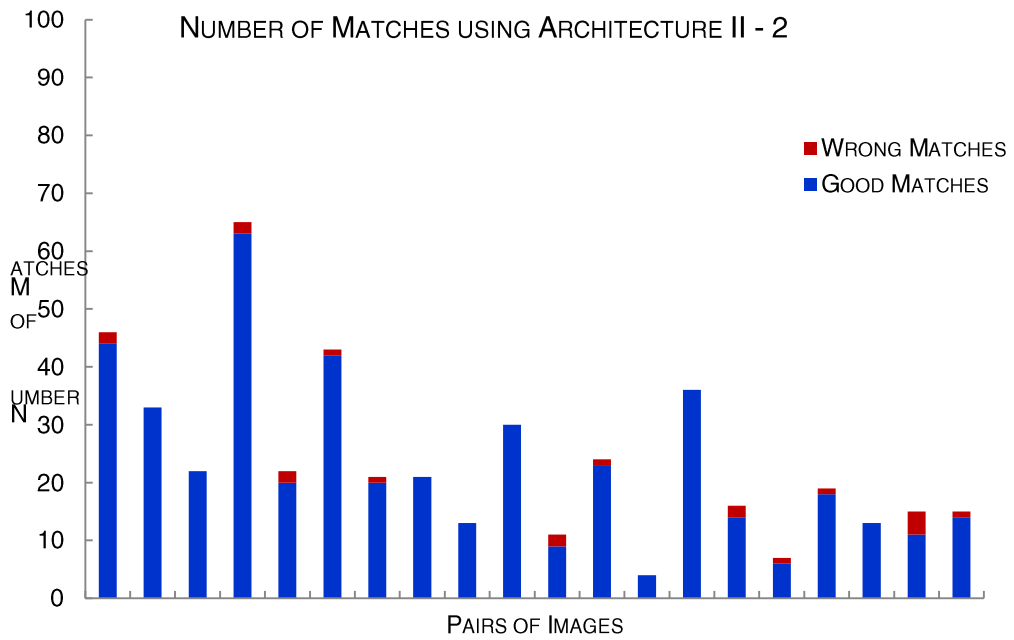


Figure 38 – Chart containing the number of matches using architecture II on the data set with perspective transformation and vessel deletion.

The mean time for computing the matches with this architecture (computed from the registration of the two datasets of segmented images) was 1,98 seconds.

5.8. DISCUSSION

From visual inspection of the actual matching of the first architecture's implementation, one can conclude that the matched key points are not uniformly distributed throughout the image, suggesting that the descriptors from the first architecture are not robust enough to account for the dissimilarities in intrinsic information of different exam modalities, which, in part, led to the implementation of the architecture with vasculature segmentation.

Architecture II performed well under the variability induced in the segmented datasets and, while further testing still needs to take place, the results are very promising. In fact, even with the visual inspection of testing performed with poorly segmented vessels from the implemented method that uses Gaussian templates, good results were obtained.

Segmentation seems to be an inevitable step for reducing heterogeneity of ocular fundus images. Therefore, the study and implementation of other approaches, like quadrature filters [28], should be a good step towards the completion of a multimodal registration algorithm.

Other considerations also have to be made regarding the transformation stage, since the second-order polynomial transformation process is not the most successful approach for registration of fundus image, as pointed out by Bernardes [29].

CHAPTER 6. CONCLUSIONS AND OUTLOOK

To conclude, this thesis contributes with a study of a brightness and contrast normalization technique that will be very helpful for ophthalmologists and for other software of pathology classification.

It also presents several improvements and modifications to the semi-assisted registration project that can be easily modified in the future to be more robust.

In addition, a good contribution to the fully-automated system was made, which was ultimately the objective of this project. The architectures presented, especially the fast SIFT with previous segmentation seem to be a very good path to the final version of such system.

In the end, a lot of questions arose along with several ideas that were discussed throughout this document. However, the time period is limited and not every idea can be implemented and tested.

Appendix A. MASK DETERMINATION

The algorithm to determine the mask created by some image acquisition systems (introduced in chapter 2) that was used throughout this project is similar to the one used by Chanwimaluang *et al.* in an implementation of the registration algorithm described in their paper “Hybrid Retinal Image Registration” [11]. This implementation has four stages (see Figure 39). In the first stage the each row of the image is analyzed from left to right and the pixels are marked as belonging to the mask until a pixel with an intensity value that is greater than 15 (in the interval $[0,255]$) is found. This value is the same as the one used by the mentioned implementation and its meaning comes from the fact that the shadow is not actually black, but a gray level close to black. When that value is found, the algorithm skips to the next line. After all lines are analyzed the second stage begins. In this stage, the lines analyzed the same way but from right to left. The last two stages are responsible of analyzing the columns, one from top to bottom and the last from bottom to top. The result is a binary map that separates mask pixels from the actual ocular fundus image.

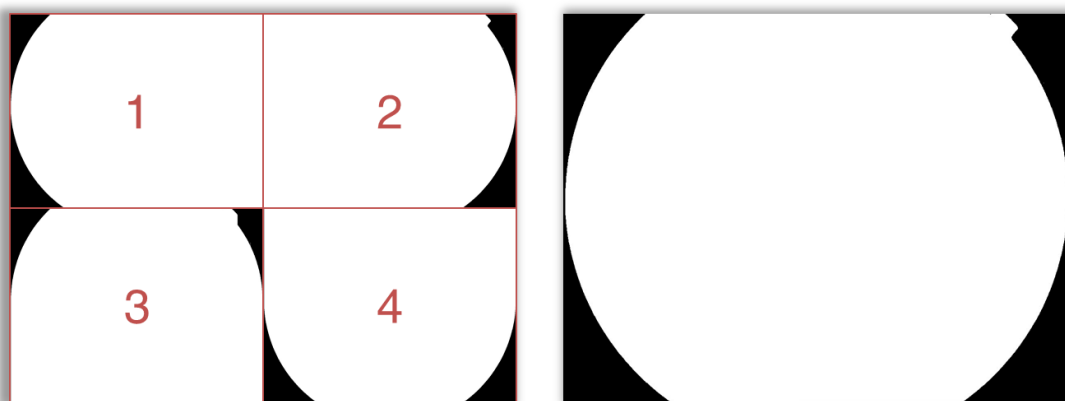


Figure 39 – Mask determination illustration.

Left: the four steps of the mask determination algorithm. Right: final mask resulting of the intersection of the intermediate steps.

Appendix B. HESSIAN MATRIX

The Hessian matrix (or Hessian) is a square matrix that contains second-order partial derivative information of a given function.

$$H(f(x_1, x_2, \dots, x_n)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (\text{B.1})$$

In a computer vision context, the Hessian typically results in a two by two matrix for each pixel $p = (x, y)$, as shown in the equation below:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (\text{B.2})$$

where L_{xx} , L_{xy} and L_{yy} result from the convolution the second-order derivatives of a Gaussian template G :

$$G(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (\text{B.3})$$

with the image itself, using the following equations:

$$\begin{aligned}L_{xx}(x, y, \sigma) &= I(x, y) * \frac{\partial^2}{\partial x^2} G(\sigma) \\L_{yy}(x, y, \sigma) &= I(x, y) * \frac{\partial^2}{\partial y^2} G(\sigma) \\L_{xy}(x, y, \sigma) &= L_{yx}(x, y, \sigma) = I(x, y) * \frac{\partial^2}{\partial x \partial y} G(\sigma)\end{aligned}\tag{B.4}$$

Appendix C. HYSTERESIS THRESHOLD

The hysteresis threshold technique is usually associated with the Canny's Edge Detector [30]. However, it can be used to binarize any given grayscale image.

Instead of using a single threshold to separate the image into two different sets of pixels, this algorithm uses two thresholds. The lower threshold is used to classify the pixels with an intensity value lower than it as the binary value "false". All pixels with intensities above the higher threshold are marked as "true" and are picked as seeds for the next stage.

At this point, the immediate neighbors of each seed are analyzed. If the neighbor pixel is greater than the lower threshold, then it is marked as "true" and it is added to the seed list; if not, the algorithm skips to the next neighbor or the next seed if there are no more neighbors. The algorithm keeps flowing until the seed list is empty.

Some pixels may have not been analyzed because they fall between the two thresholds and they do not neighbor any pixel above the higher threshold nor have a connection through pixels between the thresholds to one. These pixels are marked as "false" in a final swept through the image.

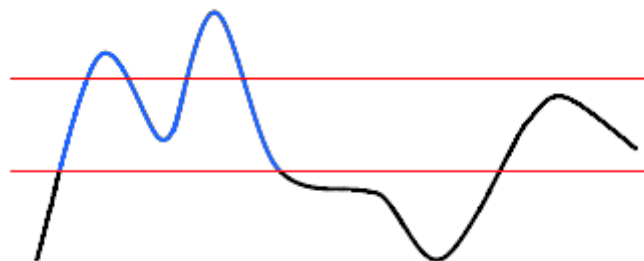


Figure 40 – Representative diagram of the application of hysteresis thresholding to a set of intensities along a row of the image. The red lines represent the two thresholds, the blue part are the pixels classified as "true" and the black part of the curve are the pixels classified as false.

Figure 40 shows the intensity values of the pixels in a row of the image and how they are classified. Obviously enough, in this representation each pixel only has two neighbors instead of the eight that are present in a two-dimensional image.

Appendix D. OTSU'S THRESHOLD

Otsu's threshold, named after its first publisher Nobuyuki Otsu [31], is a commonly used technique to perform binarization to grayscale images, splitting its histogram into two different groups of intensity values. In this method, the threshold is iteratively adjusted, changing this way the variance of both groups, with the intension of choosing the threshold that minimizes the within-class variance (σ_{wc}^2). [32]

The within-class variance can be computed with the following equation.

$$\sigma_{wc}^2 = n_B \sigma_B^2 + n_F \sigma_F^2 \quad (D.1)$$

where σ_B^2 is the variance of the first group or background, σ_F^2 is the variance of the second group or foreground and n_B and n_F are the number of pixels that belong to the first group and second group, respectively, according to the current division. After having calculated the values of σ_{wc}^2 for all possible values of the threshold, which in 8bit images can go from 0 to 255, the value of the threshold for which the within-class variance has the lowest value is chosen as the definitive threshold. [32]

Another commonly used approach uses the between-class (σ_{bc}^2) variance instead of the within-class variance, since the former is faster to compute, especially on larger images. Equation D.2 shows how it can be computed.

$$\begin{aligned} \sigma_{bc}^2 &= \sigma^2 - \sigma_{wc}^2 = n_B(\mu_B - \mu)^2 - n_F(\mu_F - \mu)^2 \\ &= n_B n_F (\mu_B - \mu_F)^2 \end{aligned} \quad (D.2)$$

where σ^2 and μ are the overall variance and mean, respectively. Then, the between-class values are searched to find the threshold that maximizes it. That threshold, like before, is set as the final threshold. [32]

Note that an additional implementation had to be made so that the determination of the Otsu's threshold in images with a black mask was not affected by the massive amount of black pixels, by previously determine the mask pixels using the algorithm described in Appendix A.

BIBLIOGRAPHY

1. Foracchia, M., E. Grisan, and A. Ruggeri, *Luminosity and contrast normalization in retinal images*. Medical Image Analysis, 2005. **9**: p. 179-190.
2. Lowe, D.G., *Distinctive Image Features from Scale-Invariant Keypoints*. 2004: Vancouver. p. 28.
3. Bay, H., et al., *Speeded-Up Robust Features (SURF)*. 2008. p. 14.
4. Junqueira, L.C. and J. Carneiro, *Histologia Básica*. 10 ed. 2004, Rio de Janeiro: Guanabara-Koogan.
5. Gonçalves, C. and V. Bairos, *Histologia - Texto e Imagens*. 2006, Coimbra: Imprensa da Universidade.
6. Lang, G.K., *Ophthalmology : a short textbook*. 2000, Stuttgart: Thieme.
7. *Human Eye*. [cited 2010 August]; Available from: http://en.wikipedia.org/wiki/Human_eye.
8. Delori, F.C. and K.P. Pflibsen, *Spectral reflectance of the human ocular fundus*. Appl Opt, 1989. **28**(6): p. 1061-1077.
9. Winder, R.J., et al., *Algorithms for digital image processing in diabetic retinopathy*. Comp Med Imaging Graph, 2009. **33**: p. 608-622.
10. Foracchia, M., E. Grisan, and A. Ruggeri, *Luminosity and contrast normalization in retinal images*. Med Image Anal, 2005. **9**(3): p. 179-90.
11. Chanwimaluang, T., G. Fan, and S.R. Fransen, *Hybrid Retinal Image Registration*. IEEE Trans Inf Technol Biomed, 2006. **10**(1): p. 129-142.
12. Gopal Datt Joshi, J.S., *Colour Retinal Image Enhancement based on Domain Knowledge*, in *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. 2008, IEEE: India. p. 591-598.
13. Lopes, F., *BW-Eye – Ophthalmologic Decision Support System based on Clinical Workflow and Data Mining Techniques – Image Registration Algorithm*, in *Faculty of Sciences and Tecnology*. 2009, University of Coimbra: Coimbra, Portugal.
14. Lowell, J., et al., *Measurement of Retinal Vessel Widths From Fundus Images Based on 2-D Modeling*. IEEE Transactions on Medical Imaging, 2004. **23**(10): p. 1196-1204.
15. Grisan, E., et al. *A new tracking system for the robust extraction of retinal vessel structure*. in *26th Annual International Conference of the IEEE EMBS*. 2004. San Francisco, CA, USA.

16. Gregson, P.H., et al., *Automated grading of Venous Beading*. Computers and Biomedical Research, 1995. **28**: p. 291-304.
17. Pedersen, L., et al., *Quantitative measurement of changes in retinal vessel diameter in ocular fundus images*. Pattern Recognition Letters, 2000. **21**: p. 1215-1223.
18. Evans, C., *Notes on the OpenSURF Library*. 2009. p. 1-25.
19. Chen, J., et al., *A novel registration method for retinal images based on local features*, in *Conf Proc IEEE Eng Med Biol Soc.* . 2008, NIH Public Access.
20. Zhu, T., *Fourier cross-sectional profile for vessel detection on retinal images*. Comp Med Imaging Graph, 2009.
21. Al-Diri, B., A. Hunter, and D. Steel, *An Active Contour Model for Segmenting and Measuring Retinal Vessels*. IEEE Trans Med Imaging, 2009. **28**(9): p. 1488-1497.
22. Stosic, T. and B.D. Stosic, *Multifractal Analysis of Human Retinal Vessels*. IEEE Trans Med Imaging, 2006. **25**(8): p. 1101-1107.
23. Chanwimaluang, T. and G. Fan, *Corrections to "Hybrid Retinal Image Registration"*. p. 1.
24. Martinez-Perez, M.E., et al., *Segmentation of blood vessels from red-free and fluorescein retinal images*. Medical Image Analysis, 2007. **11**: p. 47-61.
25. Lourenço, A.M., *Techniques for keypoint detection and matching between endoscopic images*, in *Faculty of Sciences and Technology*. 2009, University of Coimbra: Coimbra.
26. Stewart, C.V., C. Tsai, and B. Roysam, *The Dual-Bootstrap Iterative Closest Point Algorithm with Application to Retinal Image Registration*. International Journal of Computer Vision, 2002.
27. Staal, J., et al., *Ridge-Based Vessel Segmentation in Color Images of the Retina*. IEEE Trans Med Imaging, 2004. **23**(4): p. 501-509.
28. L  th  n, G., J. Jonasson, and M. Borga, *Blood vessel segmentation using multi-scale quadrature filtering*. Pattern Recognition Letters, 2010. **31**: p. 762-767.
29. Bernardes, R.M.D.C.S., *Desenvolvimento de Novas Metodologias para o Mapeamento Multimodal da M  cula*, in *Faculty of Medicine*. 2008, University of Coimbra: Coimbra.
30. Canny, J., *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **8**(6): p. 679-698.
31. Otsu, N., *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man and Cybernetics, 1979. **9**(1): p. 62-66.
32. Greensted, A. *Otsu Thresholding*. [cited 2009 November]; Available from: <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.