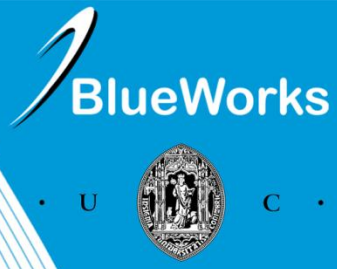


Algorithms for Ophthalmology Image Registration

Tiago Luís Leite de Bessa Ferreira

M. Sc. Dissertation

July 2012

Coimbra

University of Coimbra

Faculty of Sciences and Technology

Partnership:

Department of Physics

Department of Electrical and Computer Engineering

BlueWorks – Medical Expert Diagnosis

Algorithms for Ophthalmology Image Registration

Tiago Luís Leite de Bessa Ferreira

Coimbra 2012

Project Supervisor: Engineer Paulo Barbeiro

Project Coordinator: Professor Luís Cruz, PhD

M. Sc. Dissertation

Tiago Luís Leite de Bessa Ferreira

To obtain the degree of

Master of Science in Biomedical Engineering

Acknowledgments

First of all I would like to thank my supervisor Eng. Paulo Barbeiro for his guidance and the opportunity to develop my project within BlueWorks. I would also like to thank my coordinator Prof. Dr. Luís Cruz for his availability and helpful suggestions.
To my parents and sister a special thanks for the support since the first year of the course.
To my friends that made these 6 years in University of Coimbra a great journey, "*Thank you*".

Abstract

Image Registration is a process of aligning two or more images of the same scene taken at different times, from different viewpoints or by different sensors. The goal of this thesis is to present a manual and an automated solution to align ocular fundus images, in order to provide a valuable tool to assist the ophthalmologists in the diagnosis. In the manual side, we developed an interface where the user can select the registration points and align up to 3 images with the same reference image. The algorithm returns the available sample images aligned with the reference one. Still in the manual registration, we developed another interface to overlay structural information (angiographies and retinographies) with functional exams (perimetry reports). This solution arose as an idea discussed in a brainstorming meeting with the ophthalmologists of *Hospitais da Universidade de Coimbra*. On the automated side, we implemented the SIFT method to detect keypoints that are invariant to image scale and rotation. Then we match the keypoints by a nearest neighbour method through the Euclidian distance between the keypoints descriptors. As the polynomial transformation model is very unstable, a false positive method was implemented in order to eliminate the matches that contribute to lower the performance of the transformation. In the end of the implementation we tested the algorithm with a set of ten pairs of images where a successfully alignment in 5 of them was achieved.

Resumo

O registo de imagem é um processo para alinhar 2 ou mais imagens da mesma cena recolhidas em tempos diferentes, de pontos de vista diferentes ou por diferentes sensores. O objectivo desta tese é apresentar uma solução manual e uma automática para alinhar imagens do fundo ocular, de modo a providenciar uma ferramenta útil para ajudar os oftalmologistas no diagnóstico. Do lado manual, desenvolveu-se uma interface onde o utilizador pode escolher os pontos de registo e alinhar até 3 imagens com a mesma imagem de referência. O algoritmo devolve as imagens alinhadas com a imagem de referência. Ainda no registo manual, desenvolveu-se uma outra interface para sobrepor informação estrutural (angiografias e retinografias) com exames funcionais (relatórios de perimetria). Esta solução surgiu como uma ideia levantada numa das reuniões de *brainstorming* com os oftalmologistas dos Hospitais da Universidade de Coimbra. Do lado automático, implementou-se o método SIFT para a detecção de pontos-chave que são invariantes à escala e a rotações nas imagens. Em seguida, emparelhou-se os pontos-chave pelo método de "*nearest neighbour*" através da distância euclidiana entre os descritores destes pontos. Devido à instabilidade do modelo da transformação polinomial, um método de eliminação de falsos positivos foi implementado de forma a eliminar os pontos que contribuem para reduzir a performance da transformação. No final da implementação testou-se o algoritmo com um conjunto de dez pares de imagens onde se obteve com sucesso o alinhamento de 5 desses pares.

Contents

List of Figures

Acronyms and Abbreviations

BW	BlueWorks – Medical expert diagnosis
CCC	Centro Cirúrgico de Coimbra
DOG	Difference-Of-Gaussian
FNN	Feedforward Neural Network
GUI	Graphical User Interface
HFA	Humphrey Field Analyzer
HUC	Hospitais da Universidade de Coimbra
ICP	Iterative Closest Point
OD	Oculus Dexter
OS	Oculus Sinister
RBFNN	Radial Basis Function Neural Network
SIFT	Scale Invariant Feature Transform

Chapter 1

Theoretical Background

1.1. Motivation and Goals

This project arises from the opportunity to use technology to assist the ophthalmologists in the examination of ophthalmic images. Using the technological means currently available, it is possible to develop tools to facilitate ocular fundus image examination and thus the diagnosis of eye-related diseases.

As result of brainstorming meetings with some ophthalmologists of HUC, some interesting challenges were proposed. One of them, and the basis of this project, is the alignment of ocular fundus images. To physicians, this could be an important tool to assist them in the diagnosis or even in follow-up consults to track the evolution of certain diseases. So to answer to this need, in the first place, an algorithm and an interface window to perform manual registration of retinal images was build. Then, as an improvement of a previous algorithm developed within BlueWorks, a fully automated registration was implemented. Since BW has software (OphthalSuite) for the ophthalmology service that was designed to answer the needs of those physicians, the goal was to integrate this new tools on OphthalSuite to improve its functionalities.

Another challenge that arose in one of those meetings was the possibility to perform the registration of functional exams with ocular fundus images. This would be a combination of different modalities of ophthalmic exams that has the potential to be a valuable tool for the ophthalmologists.

1.2. The Eye

The eye is a complex organ that is composed of three layers: an external layer that consists of the sclera and the cornea; a middle layer consisting of the choroid, ciliary body and iris; and finally an inner layer of nerve tissue, the retina (figure 1).

The sclera is the opaque, white, posterior five-sixths of the external layer of the eye. It consists of tough, dense connective tissue which helps to maintain the eye shape. The remaining one-sixth anterior part, the cornea, is colourless and transparent allowing the light to pass into the eye.

At the middle layer, the choroid is a highly vascularized coat and has a high number of pigmented cells rich in melanin, which give it its characteristic black colour. Continuous to the choroid, the ciliary body is a thickened ring that lies at the inner surface of the anterior portion of the sclera. It has smooth muscle fibers called ciliary muscle that are connected to the lens. These muscles are important in visual accommodation because their contraction modify the lens shape changing its focal length. Within the ciliary body ring is the iris. It is the coloured part of the eye that is different in each person. The iris is a contractile structure with a round opening in the center, called pupil. The light enters in the eye through the pupil and the iris is responsible for controlling the amount of light that enters by shifting the pupil's size.

Finally, at the inner layer, the retina consists of two portions. A thin membrane called pigment epithelium and a photosensitive part called the neural retina. At the latter, there are millions of photoreceptive cells, the rods and cones. Examining the posterior region of retina, it's possible to distinguish some important features. Near the center of retina, there is a small yellow spot called the macula. At the center of macula exists a depression, the fovea, which is the focus point of the retina as it is the region with most photosensitive cells. The blood vessels and nerve cells come together to form the optic nerve which is the entry zone through the external layers. The optic disk has no photosensitive cells and so constitutes the blind spot of the eye. [1] [2] [3].

The ocular fundus is formed by the inner structures of the eyeball, namely, the retina, optic disk, central vasculature of the retina, macula, fovea and choroid. It is important to say that a normal retina is colourless and the red coloration seen in retinographies is result of light reflection by the choroid's vasculature. [4]. See figure 2.

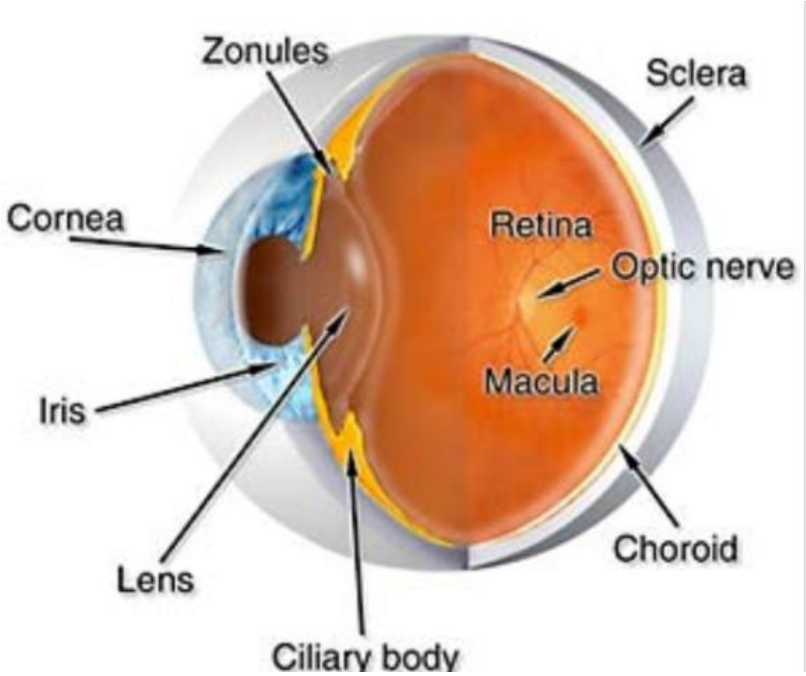


Figure 1 - Human eye anatomy [5]

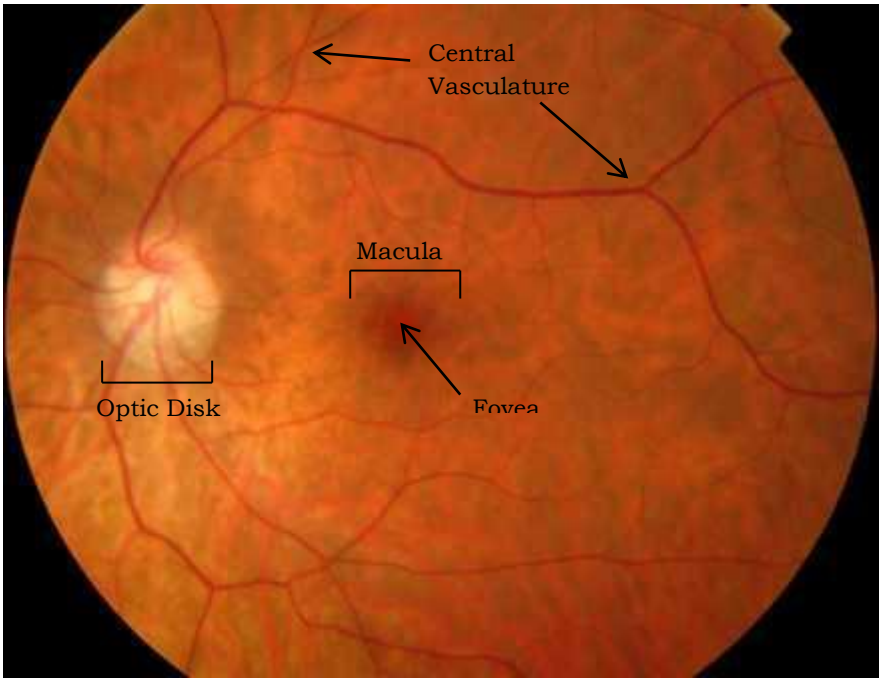


Figure 2 - Ocular Fundus Image

1.3. Retinography and Angiography

Retinography, or fundus photography, is an imaging technique to photograph the ocular fundus. This permits precise documentation of follow-up findings. Photographs obtained with a fundus camera operating mostly on the green wavelength provide high-contrast images of abnormal changes to the innermost layers of the retina, such as changes in the layer of optic nerve fibers, bleeding or microaneurysms. [4]. An example of the diseases that can be diagnosed through visual exam of the retina is glaucoma, caused by increased pressure in the inner-eye and which results in damage to the optic nerve that gets worse over time. Using serial photographs, the physician can detect subtle changes in the optic nerve caused by glaucoma and then recommend the appropriate therapy. Fundus photography is also used to help interpret fluorescein angiography because certain retinal landmarks visible in fundus photography are not visible on a fluorescein angiogram. [6].

On the other hand, fluorescein angiography is performed with a dye called *sodium fluorescein*. When illuminated with a blue light, the fluorescein dye glows or fluoresces in yellow-green. Special filters in the camera allow only the fluorescent yellow-green light to be photographed, producing high-contrast images of the retinal vessels. [7]. The most common uses of fluorescein angiography are in the diagnosis of retinal or choroid vascular diseases such as diabetic retinopathy, age-related macular degeneration, hypertensive retinopathy and vascular occlusions. Typically, these are clinical diagnosis and the angiogram is used to determine the extent of damage, to develop a treatment plan or to monitor the results of treatment. [8].

1.4. Image Registration

Image Registration, sometimes called as fusion, matching or warping, is a process of aligning two or more images of the same scene taken at different times, from different viewpoints or by different sensors. The goal is to find a transformation such that, given a reference and template image, when applied to the template makes it geometrically/ structurally similar to the reference image. This technique is applied in various fields such as biology, chemistry, physics or any area involving image processing. Specifically in medicine, image registration is used in computational anatomy, computer-aided diagnosis, fusion of different modalities, and monitoring diseases. [9]. One medical application where registration techniques are increasingly important is in automated techniques to assist in the diagnosis and treatment of diseases of the

human retina. For instance, two images taken before and after laser surgery can be registered to track the progress of diseases such as macular degeneration or glaucoma. However, retinal image registration is still a difficult issue, since several challenges must be addressed in developing reliable registration:

- Curvature of the retina surface induces errors in registration based on plane transformations,
- Illumination, which comes from outside the eye, is viewpoint dependent and cause glaring, as well as fade-outs. As a result, a region of the retina might have substantially different intensity properties in different images;
- Image overlap may be small due to large changes in viewpoint between images, thereby reducing the number of common registration points;
- Large regions of retinal images are relatively textureless. The predominant features are the blood vessel structures and, in some images, the optic disk;
- Blood vessel widths can be as narrow as two or three pixels [10].

1.4.1. Transformation

In order to overtake the aforementioned challenges, firstly it's important to understand the meaning of transformation. A transformation is a mapping of point-locations in one image to new locations in another. Transformations used to align two images may be global or local. A global one is given by a single equation which maps the entire image. On the other hand, local transformations map the image differently depending on the spatial location.
Image registration can be defined as a mapping between two images both spatially and with respect to intensity. However, the intensity transformation is not always necessary and so it will not be explained. If we define a 2D spatial-coordinate function, f, we are able to map two spatial coordinates x' and y' from the original coordinates x and y, such that:

$$(x',y') = f(x,y)$$

For a better understanding of the transformation the figure 3 represents an affine transformation scheme.

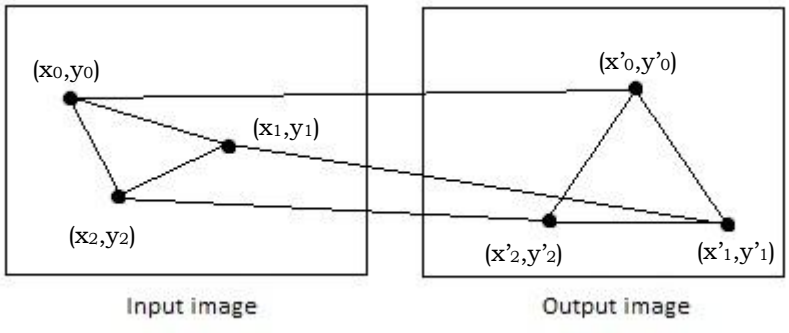


Figure 3 – Affine transformation scheme

The registration problem is to find the optimal spatial transformation so that a reference and sample images are matched to expose differences of interest between them, figure 4. On the other hand, the fundamental characteristic of any image registration technique is the type of spatial transformation or mapping used to properly overlay those two images. Although many types of variations may be presented in each image, the registration technique must select the class of transformation which will remove only the spatial distortions between images due to differences in acquisition and scene characteristics which affect acquisition. Other differences in scene characteristics that are to be exposed by overlaying should not be used to select the class of transformation.

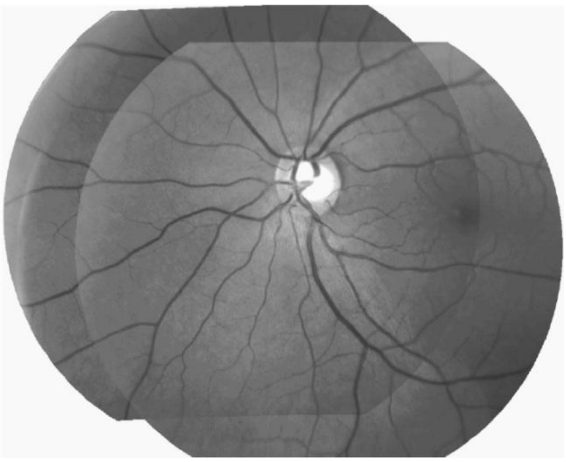


Figure 4 – Registration of Human retina images [10]

The most common general transformations are rigid, affine, projective, perspective, and global polynomial.

Rigid transformations account for object or sensor movement in which objects in the image retain their relative shape and size.

Affine transformations are more general than rigid and can therefore tolerate more complicated distortions while still maintaining some nice mathematical properties, such as its linearity. A transformation T is linear if,

$$T(x_1 + x_2) = T(x_1) + T(x_2)$$

and for every constant c,

$$cT(x) = T(cx).$$

Projective transformations and more general perspective transformations account for distortions due to the projection of objects at varying distances to the sensor onto the image plane.

Polynomial transformations are the most general global transformations (of which rigid is the simplest) and can account for many types of distortions so long as the distortions do not vary too much over the image. Unlike the polynomial transformation, the affine ones are linear in the sense that they map straight lines into straight lines. [11]. In table 1, are represented the affine and polynomial transformation models, which are the ones that were used throughout the project.

Table 1- Transformation Models [12]

Model	Transformation Models	Degree of freedom
Affine	$\begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	6
Polynomial (quadratic)	$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 & \theta_{10} & \theta_{11} & \theta_{12} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \\ 1 \end{bmatrix}$	12

1.4.2. State of the art

The majority of the registration methods consist of the following four steps:

- Feature detection. Salient and distinctive objects (edges, contours, corners, etc.) are manually or automatically detected;
- Feature matching. The correspondence between the features detected in the sensed image and those detected in the reference image is established;
- Transform model estimation. The type and parameters of a mapping function are estimated to align the sensed and reference images. The parameters are computed by means of the established feature correspondence;
- Image resampling and transformation. Using mapping functions, the sensed image is transformed. Image values at non-integer coordinate locations are computed by the appropriate interpolation technique [13].

Image registration has been the subject of intensive research in recent years, particularly towards achieving automation of the whole process. This automation is the reason for all the existing problems due to the complexity regarding the extraction of robust registration points. As it is known, the blood vessel structure is one of the most predominant features of the human retina. So, different approaches have been documented and can be generally classified into two categories: vessel-based and nonvessel-based methods.

Vessel-based methods

According to Deng [14] some unique geometric structures can be detected within the vascular trees, which can be used for feature matching. A graph-based registration framework called Graph Matching Iterative Closest Point (GM-ICP) is proposed. In the first place, the vessels are detected and represented as vascular bifurcations. Then, the ICP algorithm incorporating with quadratic transformation model is applied to register vessel shape models (figure 5). Another approach described by Bhuiyan [15], consists in the extraction of vascular features and building of a feature vector for each of the vessel segments. These vectors are represented in a tree structure containing the vascular bifurcations, branch and crossover points (figure 6). In order to match two images, this method compares the features of the same vessel in the corresponding images.

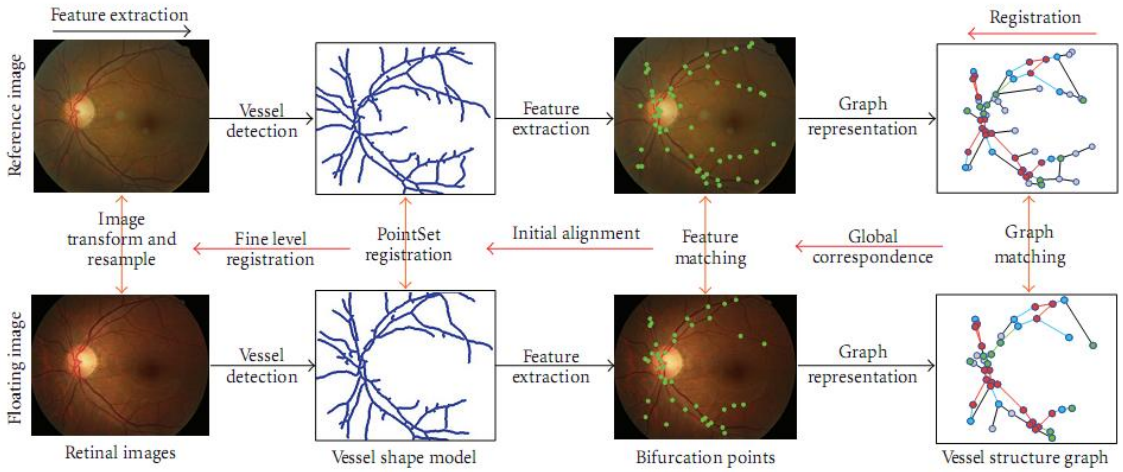


Figure 5 – Hierarchical retinal image feature extraction and registration framework [14]

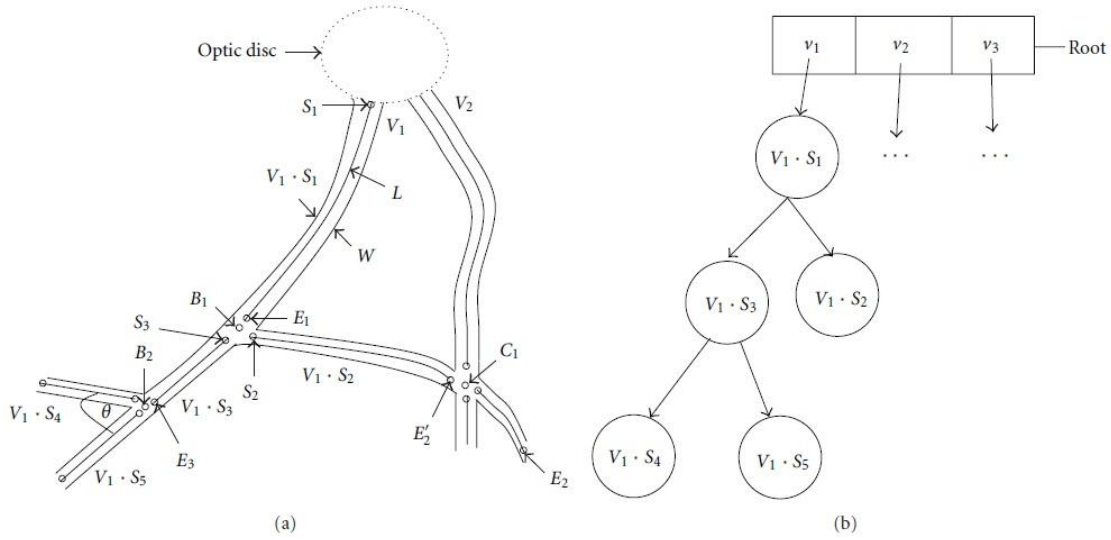


Figure 6 – A schematic diagram of a retinal image showing two blood vessels (V_1 and V_2) and the optic disc (a) and the tree model representing the binary tree for vessel V_1 (b) [15]

A different vessel-based method proposed by Chen [16], uses the bifurcation structures to match two images. The bifurcation structure is composed of a master bifurcation point and its three connected neighbours. The characteristic vector of each bifurcation structure consists of the normalized branching angle and length. It is invariant against translation, rotation, scaling, and even modest distortion. As shown in figure 7, the master bifurcation point has three surrounding branches with lengths numbered 1, 2, 3 and angles numbered 1, 5, 9, where each branch is connected to a neighbouring bifurcation point. The characteristic vector for each bifurcation structure is given by:

$$X = \{lengths, angles\}$$

The matching between vectors is made by similarity measures,

$$S_{i,j} = d(x_i, y_j)$$

Where d() is the distance measure between the vector x of an image and vector y of another image.

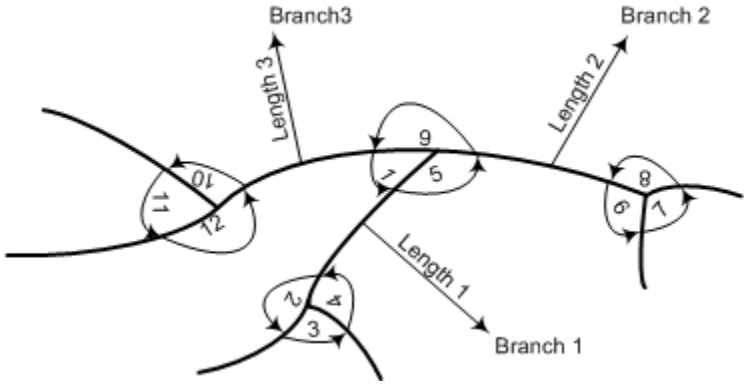


Figure 7 – Bifurcation Structure [16]

Nonvessel-based methods

In [17] Gonçalves proposes a method for automatic image registration through histogram-based image segmentation (HAIRIS). It combines some segmentations (blurring with successive filters) of the pair of images to be registered, followed by a consistent characterization of the extracted objects and a robust statistical based procedure to match objects. This methodology is schematically represented in figure 8.

In Bardera's paper [18], it is presented a framework based on compression for image registration. Two images are correctly registered when maximal compression of one image is obtained given the information in the other. The method is based on two approaches; one uses the normalized compression distance which approximates the *Kolmogorov complexity* using real-world compressors (jpeg, jpeg2000, bzip2). The other, uses the normalized entropy rate distance, which substitutes the

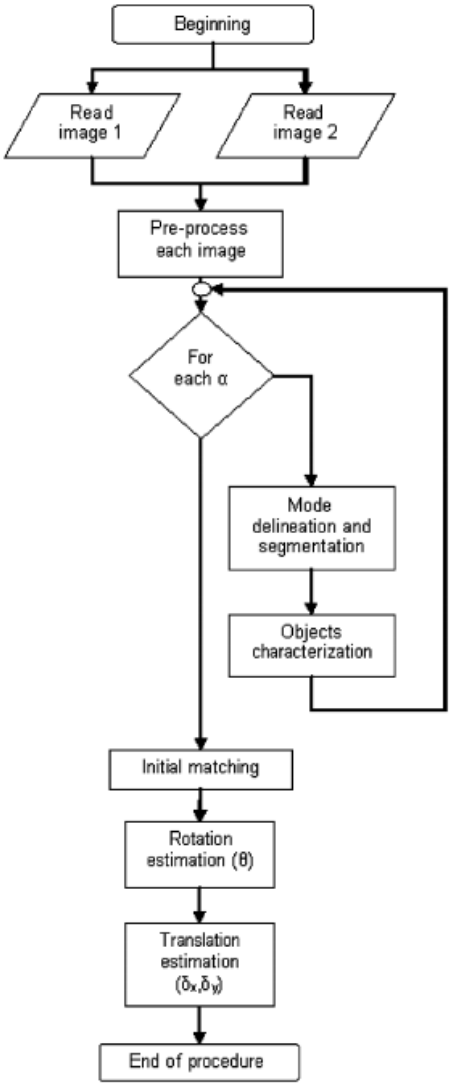


Figure 8 – Flowchart of HAIRIS method [17]

Kolmogorov complexity by the entropy rate (*Shannon version*).
A different approach to solve the automatic registration is the one suggested by Samel and Senol [19]. They propose to use a radial basis function neural network (RBFNN) instead of feedforward neural network (FNN) to find the geometrical transformation parameters. The typical neural network-based image registration is showed on the left of figure 9, while the structure of the RBFNN is presented on the right).

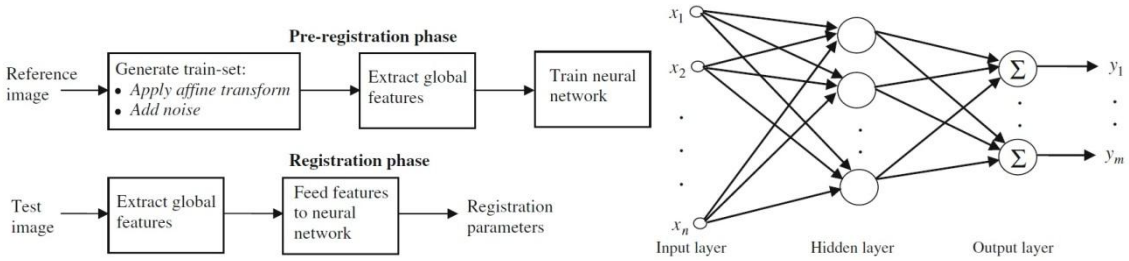


Figure 9 – Left: Neural network-based registration scheme. Right: Structure of RBFNN [19]

When a FNN is used, the training stage in the pre-registration phase is lengthy. Also, the output accuracy depends on how well the FNN has been trained. On the other hand, replacing the FNN with a RBFNN simplifies the training stage both in terms of training time and improving network generalization.

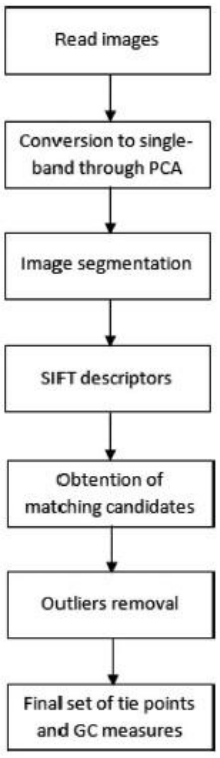


Figure 10 – Main steps of the methodology for automatic image registration [21]

Yang [20] developed a different method, the Robust Hybrid Image Matching (RHIM) algorithm, meant to optimize feature correspondence and spatial transformation and to be robust to feature extraction errors. A dynamic outlier rejection step and a local refinement technique are applied to correct the mismatched correspondences. Finally, an automatic image registration based on Scale Invariant Feature Transform (SIFT) is proposed by Gonçalves [21]. This approach combines image segmentation, SIFT method and robust outlier removal, as showed in figure 10. Therefore, an accurate set of tie points is obtained for a pair of images and the registration is performed.

Naturally, both the vessel and the non-vessel based approaches have positive points as well as disadvantages. The major disadvantage of the former is the complexity incurred to successfully achieve robust and reliable vessel detection. It's crucial to have the vascular tree correctly segmented to extract robust registration points. On the other hand, the feature-based methods can easily identify several interest points but the matching of pair of points (necessary to perform the transformation) is not easy to do reliably. This approach needs robust methods to eliminate points without match in both images, as well as remove outliers and even false matches.

Considering all the methods available as well as the goals to be achieved by this work, it was decided to try an automated registration approach based on SIFT algorithm proposed by Lowe [22]. The SIFT method is based on local descriptors assigned to each keypoint that are invariant to image scale and rotation. First, scale-space extrema detection is performed and the keypoints localization is determined. Then, each keypoint is assigned with a local orientation and a distinctive descriptor computed from the magnitude and orientation gradient of the surrounding pixels. This method and its implementation are discussed in chapter 3.

Chapter 2

Manual Registration

A registration method is considered manual when it is required a user to select the registration points. In other words, a graphical interface is needed to allow the user to select the registration points manually. These points are selected in the sample and reference images, so that they make a pair and can be used to determine the transformation model. The advantage of a manual registration over the automated one is the fact that the registration pairs are much more robust and reliable. The Downside is the need of human intervention in a time-consuming process.

2.1. Ocular Fundus Image Registration

The manual registration only became an option because of the bad results obtained with the previous automated algorithms developed within BW in former years. Since the alignment of ocular fundus images can be a valuable tool to assist the physicians, it was decided to develop and integrate this manual interface in the OphthalSuite software before trying to get better results from the automated algorithm.

2.1.1. Implemented Algorithm

The algorithm was developed using the C^{++} language and the OpenCV library. This is an algorithm that receives the reference and sample images, as well as their correspondent registration points; determines the transformation matrix; and returns two images – the aligned sample image and a pseudo-coloured image with the overlay of the reference and aligned sample. The latter is used to an easier evaluation of the

transformation accuracy (this evaluation is the users' responsibility). To a better understanding of the algorithm, it is presented next the pseudo-code and then a detailed explanation of the whole process.

Pseudo-code:

- ✓ Load reference and sample images.
- ✓ Add black pixels to both images borders.
- ✓ Create an empty image (it will be the aligned image) with size equal to the sum of the reference and sample images size.
- ✓ Calculate the transformation matrix from the registration points.
- ✓ Create the mapping images:
  - $Map_x(x,y) = \theta_1 \cdot x^2 + \theta_2 \cdot y^2 + \theta_3 \cdot xy + \theta_4 \cdot x + \theta_5 \cdot y + \theta_6$
  - $Map_y(x,y) = \theta_7 \cdot x^2 + \theta_8 \cdot y^2 + \theta_9 \cdot xy + \theta_{10} \cdot x + \theta_{11} \cdot y + \theta_{12}$
- ✓ Remap the sample image using the *cvRemap()* function.
- ✓ Create the overlap image with the size equal to the aligned image:
  - Reference grayscale image in the green channel
  - Aligned grayscale image in the blue channel
  - Red channel containing no information
- ✓ Convert the overlap image into grayscale
- ✓ Determine the cutting indexes:
  - Search from the left upper corner coordinates of reference image the first row and the first column which has the sum of all pixels equal to zero
  - Search from the right lower corner coordinates of reference image the first row and the first column which has the sum of all pixels equal to zero
- ✓ Return the aligned image; the overlap image and the cutting indexes.

As the implementation was taking place some problems arose, particularly at the mapping function. If the transformation mapped some pixels into negative indexes, it meant that the new coordinates are out of the original image size, as the left upper corner is the (0,0) coordinate. Also, the new coordinates can be mapped into indexes that are higher than the original image size. In these two situations, the *cvRemap()* function eliminates the pixels that cannot be placed within the given empty aligned image. So, to prevent the loss of valuable information, some attempts were tested. First, the *cvRemap()* code was reviewed to understand if some light modifications were enough to solve the problem. As it proved to be a failed attempt, the transformation coefficients seemed to be a better approach. The transformation matrix accounts the translation movements, so, if the coefficients responsible for translation were modified to its symmetric, maybe the mapping to

negative indexes wouldn't be a problem. Despite the success of this approach for the translation issue, the alignment of the images was not right. Once none of the other approaches worked, the solution found to circumvent the problem was to add black pixels at left and top of original images and to make the size of the aligned image bigger than the original size of the sample image.
This may not be the best approach to solve the problem, but is one that works and gives the sample image enough freedom to significant translation. So to the original reference and sample images are added two bands of black pixels at the left and top of the images, figure 11. On the other hand, the aligned image is created to have the size equal to the sum of reference and sample images size, figure 12.

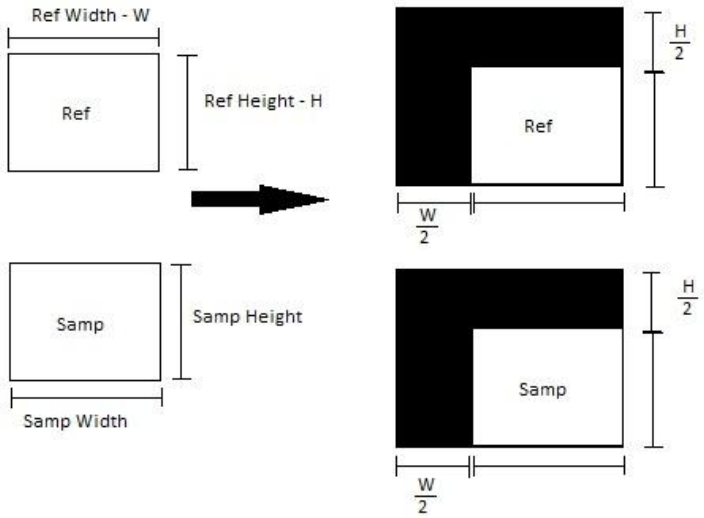


Figure 11 – Representation scheme of the addition of black pixels to reference and sample images

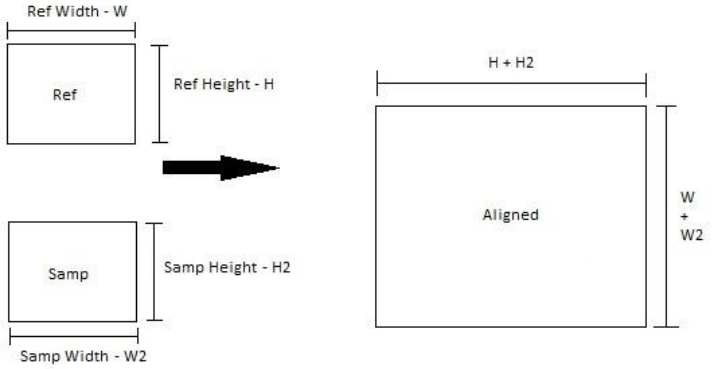


Figure 12 – Representation scheme of the empty aligned image.

The next step is the calculus of the transformation matrix from the registration points. Accordingly to table 1, the polynomial quadratic model is given by:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 & \theta_{10} & \theta_{11} & \theta_{12} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \\ 1 \end{bmatrix}$$

To be possible to solve the above equation, at least 12 points are needed, i.e. 6 points in each image that are pairs and correspond to the same features in both images. With the keypoints of the sample image two vectors are created,

$$U = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]'$$
$$V = [y_1\ y_2\ y_3\ y_4\ y_5\ y_6]'$$

On the other hand, with the keypoints from the reference image the next matrix is built,

$$D = \begin{bmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2y_2 & x_2 & y_2 & 1 \\ & & \cdots & & & \\ x_6^2 & y_6^2 & x_6y_6 & x_6 & y_6 & 1 \end{bmatrix}$$

Now we are able to determine the transformation matrix for xx coordinates (θ_x) as well as the transformation matrix for yy coordinates (θ_y), by performing the next two equations:

$$\theta_x = D^{-1} \cdot U$$
$$\theta_y = D^{-1} \cdot V$$

Two mapping images (Map_x and Map_y) need to be created since they will be responsible for the correct "filling" of the aligned image at the remap stage. As showed before in the pseudo-code, these images are created according to

$$Map_x(x,y) = \theta_1 \cdot x^2 + \theta_2 \cdot y^2 + \theta_3 \cdot xy + \theta_4 \cdot x + \theta_5 \cdot y + \theta_6$$
$$Map_y(x,y) = \theta_7 \cdot x^2 + \theta_8 \cdot y^2 + \theta_9 \cdot xy + \theta_{10} \cdot x + \theta_{11} \cdot y + \theta_{12}$$

Where *x* and *y* are the coordinates of each pixel of the aligned image.
Then, the *cvRemap()* function fills the previously created aligned image performing,

$$Aligned(x,y) = Samp(Map_x(x,y), Map_y(x,y))$$

Once there was the need to add the black pixels earlier, so relevant pixels weren't lost, they can now be eliminated. In order to achieve that, the reference and aligned images are overlaid and the cutting indexes are determined. The overlay is made by converting the reference and sample images to grayscale (only if necessary), placing the reference image in the green channel, the sample image at the blue one, and the red channel is completed with a black image (image of zeros) For a better understanding of the process, figure 13 is a schematic representation of the coordinates of the cutting indexes. In order to find those indexes, the overlap image is converted to grayscale and then the pixels along a column and a row are summed. This way, if a line is completely black, the sum is returned zero and it means that can be deleted. Since the size and coordinates where the reference image is placed are kown, it becomes faster to find the cutting indexes once that the search begins at the left upper corner and the right lower corner of the reference.

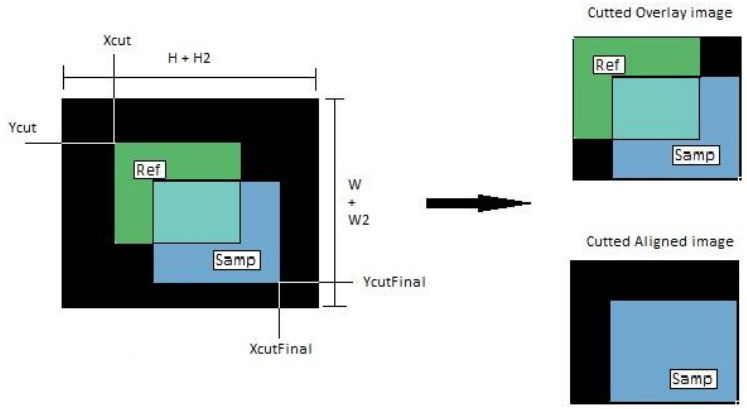


Figure 13 – Representation scheme of the cutting indexes

The algorithm returns the uncut overlap image (so the user can preview the results of the alignment), the uncut aligned image and the cutting indexes. There are two reasons why the images are not cropped within the algorithm: First, the memory management necessary because of the interoperability of C# and C++ environment, i.e. images are passed between them as pointers to the memory, so to receive an image returned by C++ algorithm, it has to be previous declared in C# with the respective size and format; The second reason is the possibility to align multiple images. In other words, if the user chooses to align more than one image with the reference, at the end, all the images must be aligned between them. As there is the need to add the black pixels, so no information is lost, we will have cutting indexes specific to each image.

In order to align all images, we must choose the indexes that cover a greater area, so the images will be cropped all by the same.

2.1.2. Graphical User Interface

The graphical user interface (GUI) was developed in C# language, and the code was adapted to use the C++ transformation algorithm, and is integrated in OphthalSuite software. The GUI is composed by two distinct parts: the first one was projected for the user to choose the images to align and set which one of them is the reference (*GUI_Selection*); the second part was developed so the user sets the registration pairs and performs the alignment (*GUI_Alignment*). Throughout this sub-chapter, both parts of the GUI, as well as their multiple options, will be presented and properly described.
To start, figure 14 shows the first part of the GUI where the user chooses the images to align and which of them is the reference. The images are organized by the eye that they represent, OS for left eye and OD for the right eye. Naturally, it only makes sense to align images that are of the same eye. Although the user can select as much images as he wants for the alignment on OphthalSuite, at *GUI_Selection* part, only four images per eye (at most) can be passed to the *GUI_Alignement*. There are two major considerations to be aware at the *GUI_Selection*:

- Number of images. As said before, up to 8 images (4 for each eye) can be sent to the *GUI_Alignment* part. If there are more than 4 images in a set, a red warning appears on the top of it, as it is visible in figure 14. Same thing happens if a given eye has only one image, the user is warned that the image will be discarded. This situation has a simple explanation, which is the need of a reference and sample images to perform the alignment. In case of both the eyes have just one image each, the GUI is closed and gets back to the main window of OphthalSuite.
- Reference image. In order to perform the alignment, there is a need to set a reference image. This one is the image that will remain unchanged and the goal is to align the other images according to the reference. The selection of the reference is thus mandatory for each eye with more than one image imported into the GUI. Figure 15, shows the selected reference images.

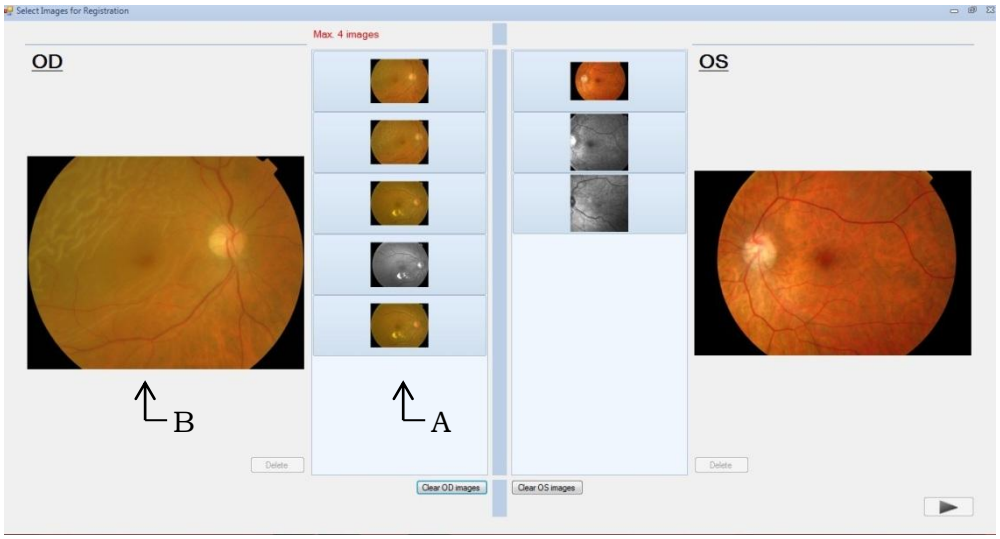


Figure 14 – First part of the GUI ("Select images for registration") with excess of OD images

When the user left-clicks in one of the miniatures (figure 14 – A), first it is set as the reference image and in the second place the correspondent picturebox (figure 14 – B) shows the enlarged image to a better visualization. If the user wants to delete images, there are two options: delete all images of one eye at once; delete only the selected image. The delete buttons are placed in the correspondent side. "Clear OD images" button (figure 15 – A) deletes all OD images at once and the "Delete" button (figure 15 – B) is only enable if an image is selected. The right-arrow button (figure 15 – C) on the right lower corner of the GUI window is the advance button. To advance for the *GUI_Alignment* part, the user presses the button and if any of the conditions (number of images and references) is not properly set, a warning pops-up.

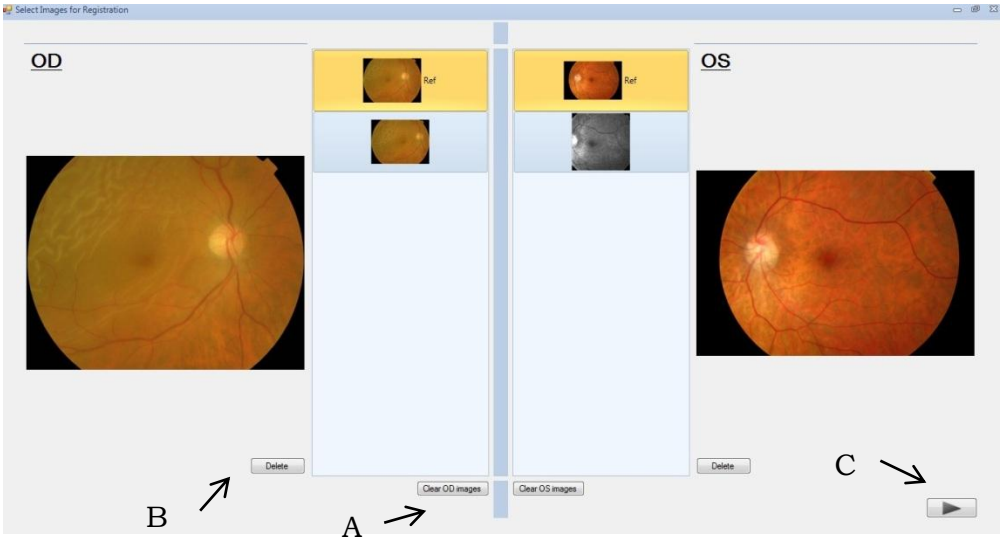


Figure 15 – First part of the GUI ("Select images for registration") with references set

Advancing to the second part of the GUI (*GUI_Alignment*), the "Image Registration" window, the images are also organized by the eye that they represent, figure 16. Using the buttons ("OD"/"OS") on the top of the window (figure 16 – A), it is possible to navigate between the right and left eye images. The reference image is placed in the left picturebox and the sample image on the right one. In case more than one sample image for a specific eye exists, the top right buttons (figure 16 – B) are set as enable, and it is possible to swop between the sample images of the correspondent eye. Still in the top of the window, on the left, there are two buttons (figure 16 – C): anchor images and reset position. When selected, the former makes possible to drag both images at the same time, i.e. if the user drags one of the images (using the left mouse button) the other moves accordingly. Reset position button in turn, places both images to their initial position. In other words, makes the zero coordinate of the image (left upper corner) correspond to the zero coordinate of the picturebox. These two buttons are only needed when the original image size is larger than the picturebox size, once the image is not rescaled to ease the accurate registration point position determination.

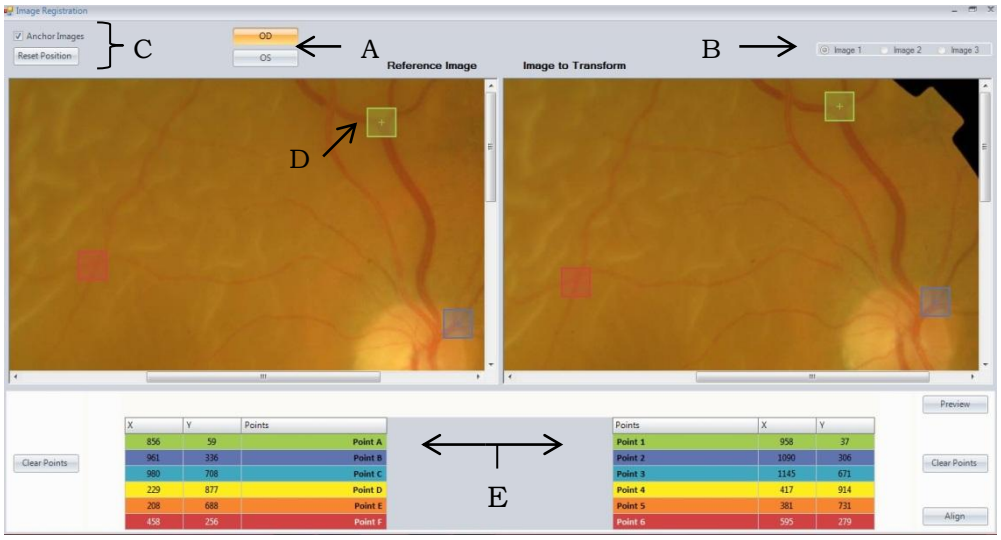


Figure 16 – Second part of the GUI ("Image Registration") OD points selected

The major goal of the *GUI_Alignment* part is to enable the manual selection of the registration points. To select the points, the user needs to right-click on the desire point on the image, and a coloured square (figure 16 – D) is centred at the selected coordinates. This coordinates are presented on the list below the image (figure 16 – E) with the same colour of the correspondent point. The colour code has as its main purpose the matching of the points of both images. For instance, as it is represented in figure 17, the orange square on the left, corresponds to

the same structure of the orange square on the right image. It is crucial that the points in both images with the same colour correspond to the exact same structure, so the transformation can be performed correctly. The polynomial quadratic transformation is very sensitive to the precise position, at the pixel level, of the two points of a match. Slight differences and errors between the keypoints position can be overspread and result in a bad alignment. This unpredictable behaviour, also can be minimized if the registration points are as far as possible from each other so they can cover a greater area of the image. If the mouse pointer is placed within a coloured square, using the left button, the user can drag it and the keypoint new position is refreshed. On the other hand, the user can delete all the selected points, of the current image, pushing the "Clear Points" button placed below the correspondent image. The selected points are specific to each image, i.e. the GUI stores in memory the selected points for the sample and the correspondent points of the reference image and when swapping between the available images, the previously stored points for the current image are presented. Notice that, when the "Clear Points" button is pressed, only the points of the current image are deleted.

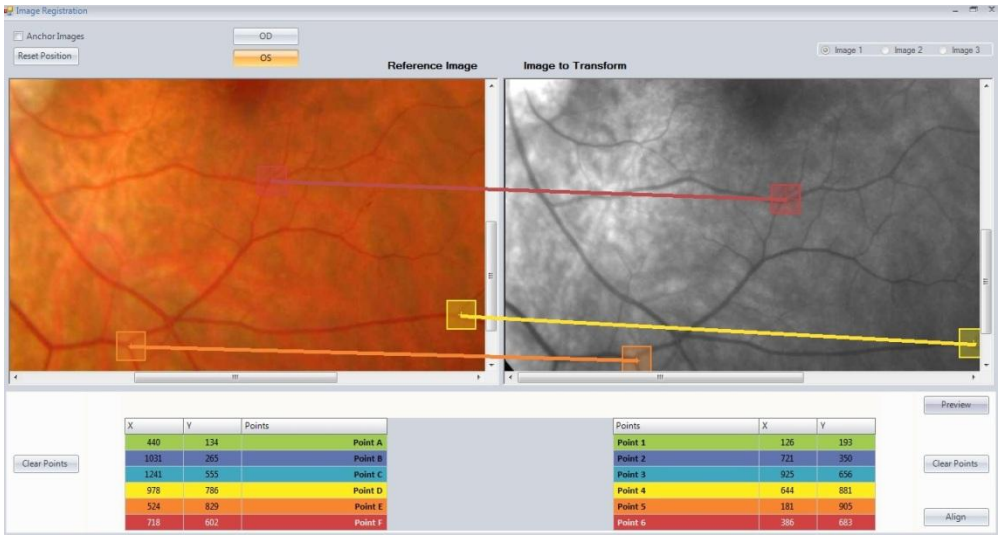


Figure 17 – Second part of the GUI ("Image Registration") OS points matches

As we can see in both figures 16 and 17, placed at the left side on the bottom of this part of the GUI, there are two buttons. The "Preview" button, applies the transformation algorithm to the two currently presented images. The "Align" button in turn, applies the transformation algorithm to all the available images and exits the GUI to return to OphthalSuite window with the aligned images. However, in both cases, each image has to compulsorily have six selected registration points. As represented in figure 13, the "Preview" button

will show a cropped overlay image, so the user can see the result of the transformation in order to decide if it's good enough, or if there is the need to adjust some of the keypoints. The presented image has a pseudo-colour, where the green channel represents the reference image and the blue channel represents the aligned sample image, figure 18. On the other hand, the "Align" button performs the transformation to each pair of images available (reference and sample), and returns the aligned sample images as well as their cutting indexes. Then, these indexes are compared (within each eye) and those who will set a greater area are selected to crop all the images of the correspondent eye. In other words, to the left upper corner the lower indexes are selected while the higher indexes are selected for the right lower corner. Naturally, the more images there are to transform the more time will be needed to complete the alignment.

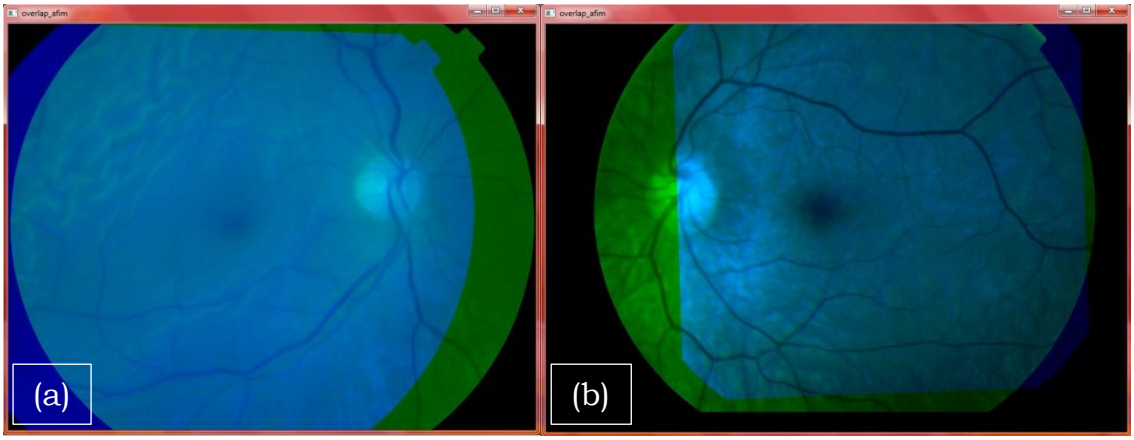


Figure 18 – Resultant overlay images of "Preview": (a) OD eye; (b) OS eye

2.2. Perimetry Exam Registration

Perimetry, or visual field testing, is a diagnostic test procedure that is commonly used to detect, diagnose, and follow-up many ocular and neurologic diseases. Nowadays, this procedure is done by automated perimeters as the Fieldmaster, Humphrey Field Analyzer (HFA), Humphrey Matrix, Octopus, Easyfield, and Medmont, [23]. For this project, *Centro Cirúrgico de Coimbra*, kindly allowed us some HFA anonym reports as well as the ocular fundus images of the correspondent patients.
Humphrey Field Analyzer consists of a hemispherical bowl onto which a target can be projected at any location in the usual field. A HFA report has several components [24]:

- The numerical display (numerical grid) is located to the left of the grey scale and to the right of the reliability indices (figure 19).
- The grey scale represents the adjacent numerical display in graphical form and is the simplest display to interpret. Decreasing sensitivity is represented by darker tones (figure 19).
- Reliability indices reflect the extent to which the patient's results are reliable and should be analysed first. If grossly unreliable, further analysis of a visual field printout is of little value (figure 19).
- Total deviation display represents the difference between the test-derived threshold at each point and the normal sensitivity at that point in the general population, corrected for age (figure 20).
- Pattern deviation is derived from the total deviation values adjusted for any generalized decrease in sensitivity in the overall field (figure 20).
- Probability displays are located below the numerical total and pattern deviation displays. These constitute a graphical representation of the percentage of the normal population in whom the measured defect at each point would be expected. Darker symbols represent a greater likelihood that a defect is significant (figure 20). [24]

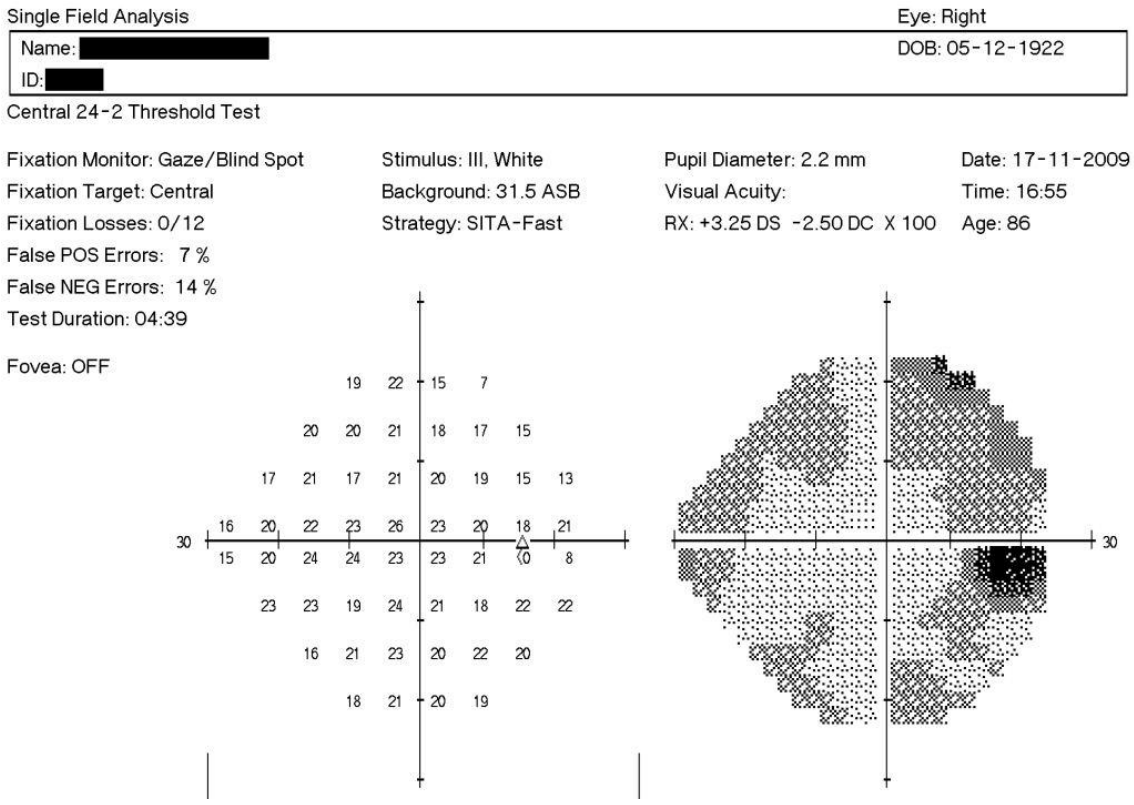


Figure 19 – HFA report (top), provided by CCC

During this project, a brainstorm with Ophthalmologists was performed to better guide research lines. One of the proposals made was to enable the assessment of functional and structural information by registering fundus images to results of perimetry exams.
As the grey scale display is the simplest to interpret, our approach for the perimetry exam registration, was to overlay the grey scale graphical display with the correspondent patient's ocular fundus image.

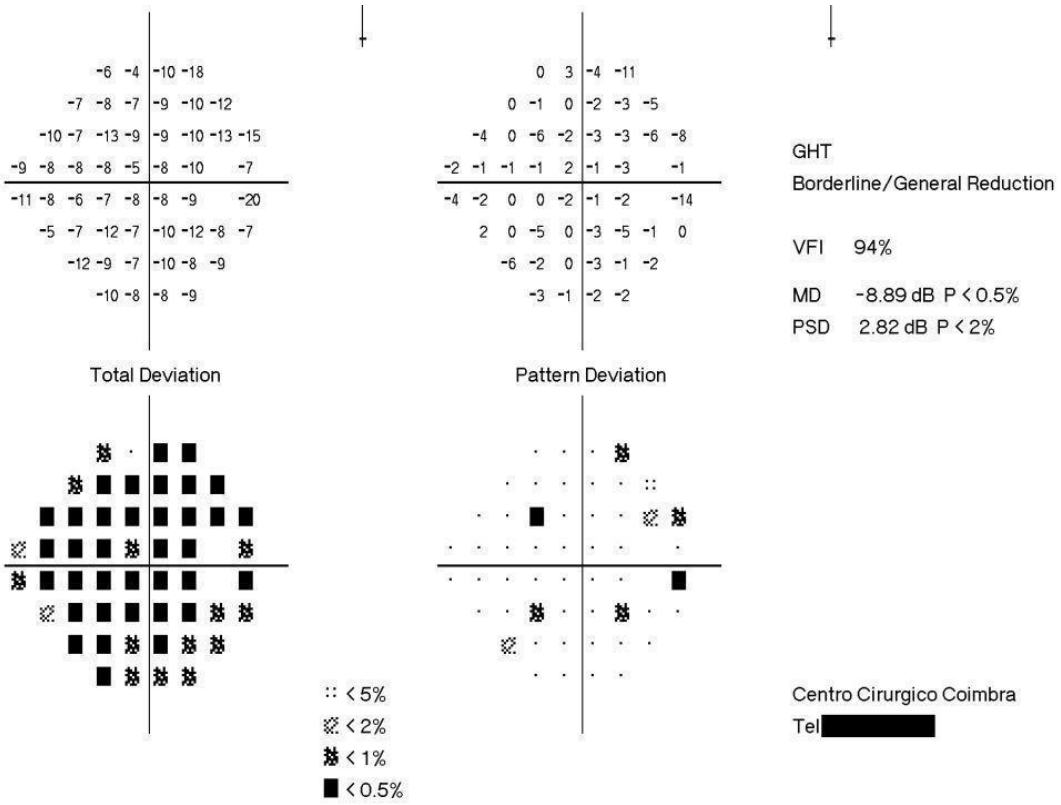


Figure 20 – HFA report (bottom), provided by CCC

2.2.1. Implemented Algorithm

As said, our approach was to overlay the grey scale display with the ocular fundus image. In order to do that, we build an algorithm using the C++ language and the OpenCV tools. The algorithm receives from the GUI, the fundus image, the grey scale display image and 3 pairs of registration points. It is very similar to the ocular fundus registration algorithm, but different from the transformation model. Since the functional information doesn't have structural information, alignment information will be extracted from two references: central position and blind spot. Due to this low detail, a simple affine transformation will be enough.

The pseudo-code is now presented to ease the understanding of the proposed method.

Pseudo-code:

- ✓ Load fundus image (as reference image).
- ✓ Load HFA grey scale display (as sample image).
- ✓ Create an empty image (it will be the aligned image) with the size equal to the max size of both loaded images.
- ✓ Calculate the transformation matrix from the registration points.
- ✓ Create the mapping images:
  - $Map_x(x,y) = \theta_1 \cdot x + \theta_2 \cdot y + \theta_3$
  - $Map_y(x,y) = \theta_4 \cdot x + \theta_5 \cdot y + \theta_6$
- ✓ Remap the sample image using the *cvRemap()* function
- ✓ Create an alpha image to the transparency level of each pixel of the aligned image.
- ✓ Merge the aligned image with the alpha channel.
- ✓ Return the final aligned image.

After loading the reference fundus image and the sample image (grey scale display) the transformation matrix is determined from the pairs of keypoints. As presented in table 1, the affine transformation model is given by:

$$\begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \text{Eq(2)}$$

In order to solve equation Eq(2), at least 6 points are needed, i.e. 3 points in each image that are pairs and correspond to the same structures in both images. With the keypoints of the sample image two vectors are created,

$$U = [x_1\ x_2\ x_3]'$$
$$V = [y_1\ y_2\ y_3]'$$

On the other hand, with the keypoints from the reference image the following array is build,

$$D = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

Now we are able to determine the transformation matrix for xx coordinates (θ_x) as well as the transformation matrix for yy coordinates (θ_y), by performing the next two equations:

$$\theta_x = D^{-1} \cdot U$$
$$\theta_y = D^{-1} \cdot V$$

Two mapping images (Map_x and Map_y) need to be created since they will be responsible for the correct "filling" of the aligned image at the remap stage. As showed before in the pseudo-code, these images are created according to

$$Map_x(x,y) = \theta_1 \cdot x + \theta_2 \cdot y + \theta_3$$
$$Map_y(x,y) = \theta_4 \cdot x + \theta_5 \cdot y + \theta_6$$

Where x and y are the coordinates of each pixel of the aligned image.
Then, the *cvRemap()* function fills the previously created aligned image performing,

$$Aligned(x,y) = Samp(Map_x(x,y),Map_y(x,y))$$

The pixels in the aligned image that weren't mapped by the above equation are filled with zero (black pixel). In order to give transparency to the aligned image, the fourth channel must have the levels of transparency of each pixel. This channel is completed by creating and merging the alpha image, which will set the lighter pixels to be 100% transparent, and the darker pixels to be 50% transparent. This way, the white background of the grey scale display is not visible. The algorithm returns the aligned transparent grey scale display.

2.2.2. Graphical User Interface

The graphical user interface (GUI) was developed in C# language adapted to use the C++ transformation algorithm, and is a "stand alone" tool. This tool has as its main purpose the overlay of functional information (HFA report) with structural images (ocular fundus images). Once this software is still a first prototype, the developed GUI is prepared to load the images, manually select the registration points and visualize the resultant overlapped image. It is quite alike to the second part of the ocular fundus registration GUI, however has some significant differences in the following features:

- Number of images. The user can load two images (reference and sample), and does not have the possibility to align more than one sample image at once.
- Number of keypoints. Since it is used an affine transformation model, it is only needed to have 3 matched pairs of points. However, as it will be explained later, the user just has to select two points in each image and the third ones are geometrically determined.
- Overlay image. Unlike the manual registration, this overlay image is not represented with a pseudo-colour but instead by the transparent grey scale display on top of the original fundus image.

As showed in figure 21, for the user to load the images, below the respective picturebox there is a “Load” button. When the left load button is pressed, the user must choose an ocular fundus image and then a pop-up question appears to define the images as left eye (OS) or right eye (OD), figure 22. This is a crucial element of the GUI because it influences the third keypoint determination. Automation could be achieved by detection of optical nerve location and vessel arcades, but that was outside the scope of this project. On the other hand, pressing the right load button, the user chooses a HFA report of the same patient. However, as the grey scale display is the simplest to interpret, it is the only display that we want to align and so the only one that should be presented in the right picturebox. To do so, as we know that all reports have the same size and display distribution, we know the exact position where the grey scale display is, and this way, the report is cropped by the display’s area. This may not be the better way to automatically obtain the grey scale display, however it is enough for a first software prototype.

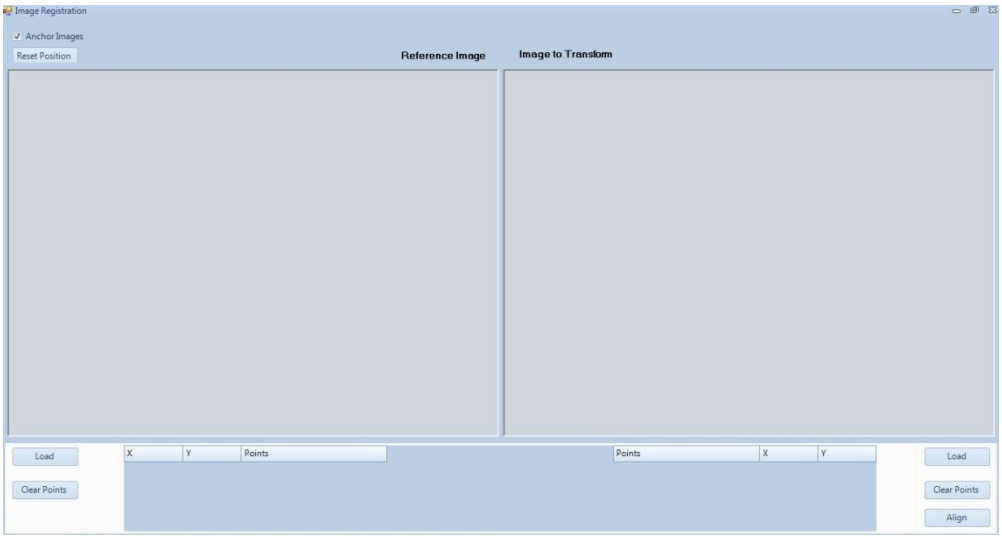


Figure 21 – Perimetry GUI main window

When both images are loaded, the user must select the registration points. As said before, it is required 3 matched pairs (6 registration points) to solve the affine transformation model. However, the user will choose only 2 pairs (2 points in each image) for a simple reason, the structural information of the grey scale display. As presented earlier, the displays in the HFA reports have functional information, but they don't have the structural data needed to select the keypoints. The only thing we know in the display is that the interception of the axis (zero coordinates) corresponds to the fovea and the darker circle-like area is the blind spot (which corresponds to the optic disc).

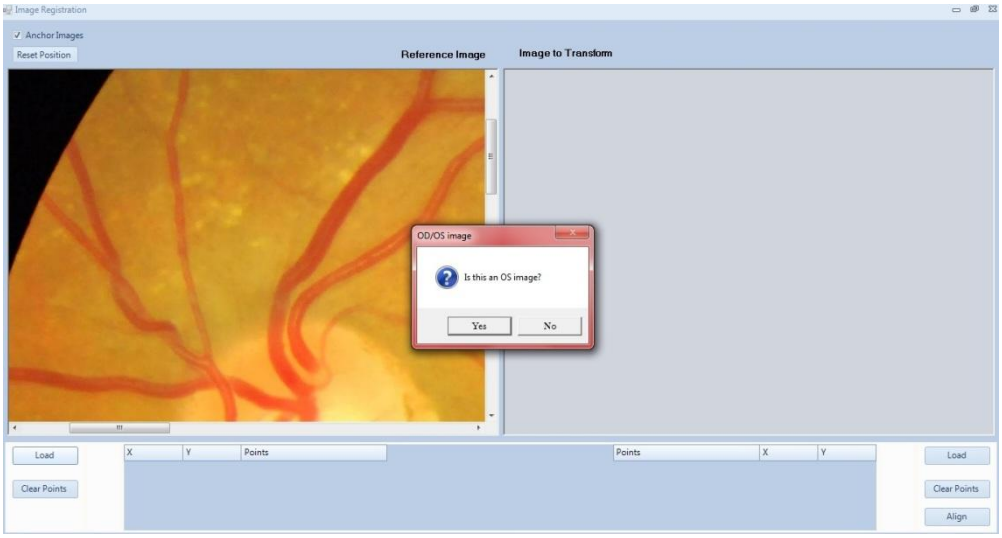


Figure 22 – Perimetry GUI OS/OD pop-up message

As represented in figure 23, the user has to mark the fovea and optic disc centres in the fundus image as well as the interception of the axis and the blind spot centre on the perimetry display. Notice that, as in the ocular fundus registration, the colours are meant to match the points in both images. Besides that, all the other buttons (excluding the "Align" button) have the same functions, for instance, to mark a keypoint the user has to right-click on the position and to drag the images just keep the left mouse button pressed and move to the desired position. Also, the keypoints coordinates are registered on the grid below the images, and if they are dragged, their new coordinates will be automatically refreshed.

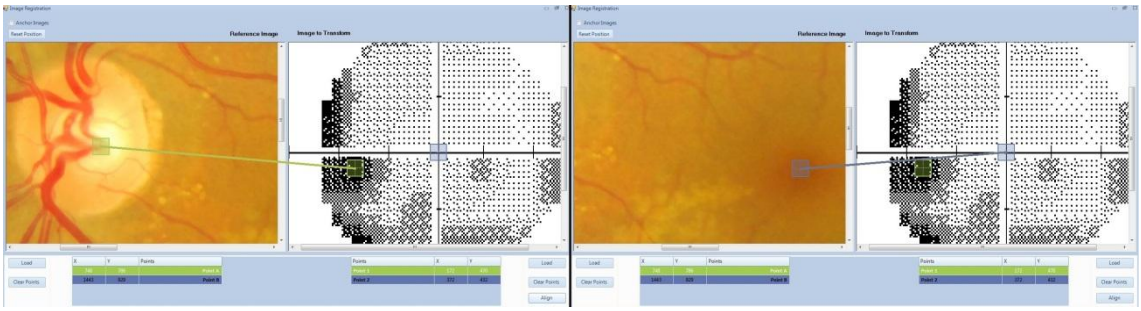


Figure 23 – Perimetry GUI. Left: the optic disc keypoint. Right: the fovea keypoint

Once it is required 3 matched pairs, and only two of them can be selected by the user, we had to come up with a solution in order to solve the affine transformation. We propose a geometric relation to determine the third keypoint, based on right triangles. Assuming that the fundus image does not have great deformations, it is possible to determine a third point that forms a right triangle with the fovea and optic disc points. If we do the same in the perimetry display, and if the distance from the optic disc to fovea is the same from the fovea to the third point (within each image), then the two triangles are geometrically related. Figure 24 is a representation scheme of the proposed approach for the third pair determination.

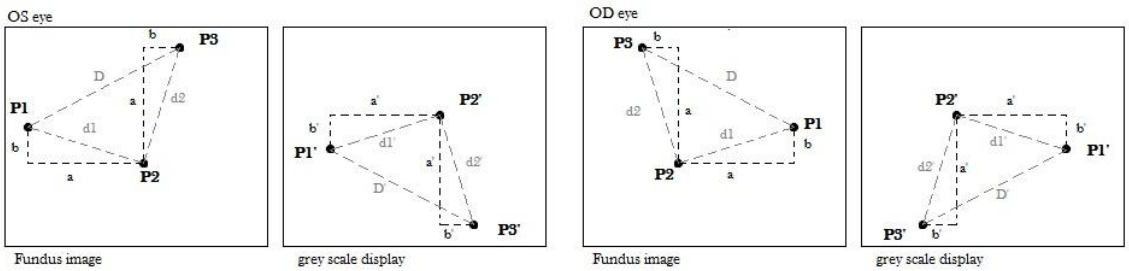


Figure 24 – Representation scheme of the third keypoint position determination

P1 and P2 represent the selected keypoints that correspond to the optic disc centre and the fovea, respectively. Similarly, P1' and P2' are the keypoints for the perimetry display that correspond to the blind spot centre and the axis interception, respectively. The distance between the optic disc and the fovea, d1, is determined from the distance between them by the xx coordinate, a, and by the yy coordinate, b. Assuming that the distance from P3 to the fovea, d2, is equal to d1, we can calculate the distance D between P3 and the optic disc and so determine the exact position of P3, which forms a right triangle. If we do the same calculations for the perimetry display, the resultant right triangle will be similar to the one determined for the reference fundus image. However, as said before, we know that the grey scale display is vertically flipped, so if we choose the P3' (third point for the perimetry) in the opposite direction, and so the affine transformation will align both images correctly.

As it is perceptible, marking the optic disc and the blind spot centres is not a very precise task. A little shift in the marked keypoints can cause too much (or too less) rotation and scaling to the aligned perimetry display. To compensate that, when the overlay image is showed there are two options on the window top at the left, see figure 25. The first one is a scale tool that allows the user to increase (or decrease) the grey scale display size up to 15%. The second one in turn is a rotation tool that allows a rotation up to 45 degrees both to clockwise and contraclockwise direction. It is important to say that the rotation tool has as fixation point the fovea, i.e. the axis interception is fixed and the grey sale display is rotated according to this point.

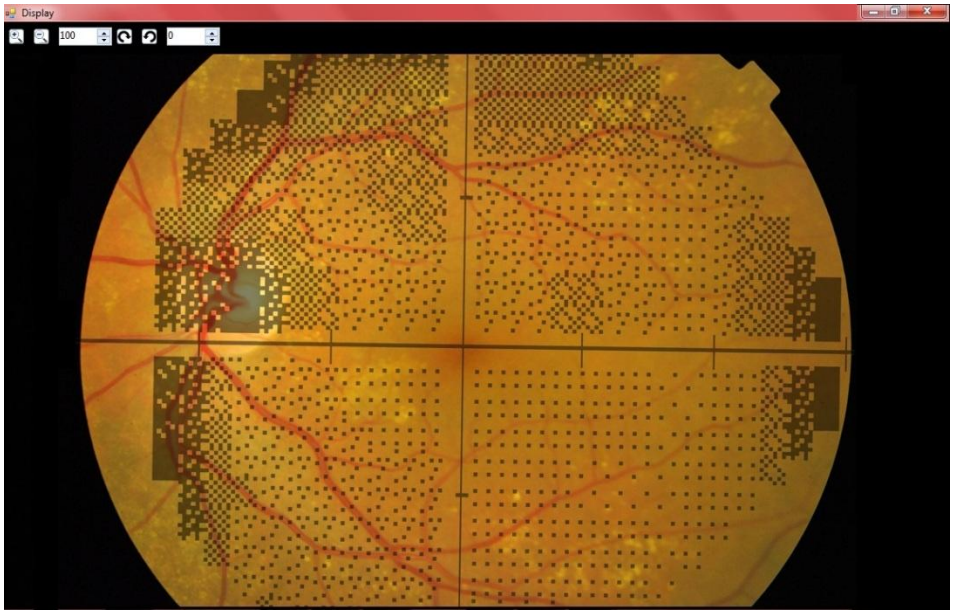


Figure 25 – "Display" window with the overlay image of ocular fundus image and perimetry information

Chapter 3

Automatic Registration

As said before, automated methods for image registration have been the subject of intensive research in the recent years. In the medical imaging field, some advances where accomplished due to these techniques, in order to improve the diagnosis processes.
The second main goal of this project was to achieve a fully automated registration method for ophthalmic images. We initially tried to implement a multimodal automated registration, but as the results were starting to show up, we decide to keep the method unimodal in order to optimize the implementation and get a fully functional prototype.
In the state of the art (chapter 1) some approaches are mentioned, but the one that we thought to better fit our necessities was the SIFT method proposed by Lowe [22]. He proposes an approach based on the four following steps: Scale-space extrema detection; Keypoint localization; Orientation assignment; Keypoint descriptor. The features extracted are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. Next, it will be presented the diverse stages of the algorithm to extract the keypoints, as well as the matching and false positive methods implemented.

3.1. Scale-space extrema detection

The first method is responsible to detect the location of all possible keypoints, which are invariant to scale and space. It identifies locations and scales that can be assigned under differing views of the same object. Furthermore, the searching for stable features across all possible scales allows the detection of locations that are invariant to scale [22]. These locations are selected at maxima and minima of a Difference-Of-Gaussian (DOG) function applied in scale-space [25].
The scale-space of an image is defined as a function, $L(x,y,\sigma)$ that is the result of the convolution of a Gaussian function, $G(x,y,\sigma)$, with the input image $I(x,y)$.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Where '*' is the convolution operation in x and y. The difference-of-gaussian is computed from the difference of two nearby scales separated by a constant multiplicative factor k [22]:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)$$

The goal is to build a DOG pyramid, in order to search for the extrema across all scales. So, to build the pyramid, the input image is first convolved with the gaussian function using $\sigma=\sqrt{2}$ to give an image A. This is then repeated a second time with a further incremental smoothing of $\sigma=\sqrt{2}$ to give a new image, B, which now has an effective smoothing of $\sigma=2$. The difference-of-gaussian function is obtained by subtracting image B from A. Figure 26 is a schematic representation of the building of the DOG pyramid. Moreover, the initial image is incrementally convolved with gaussians to produce images separated by a constant factor k in scale space ($k=\sqrt{2}$), to create an octave. To generate the next pyramid level, the gaussian image that has twice the initial value of σ is resampled using bilinear interpolation with a pixel spacing of 1,5 in each direction. The 1,5 spacing means that each new sample will be a constant linear combination of the four adjacent pixels. This is efficient to compute and minimizes aliasing artefacts that would arise from changing the resample coefficients [22] [25].

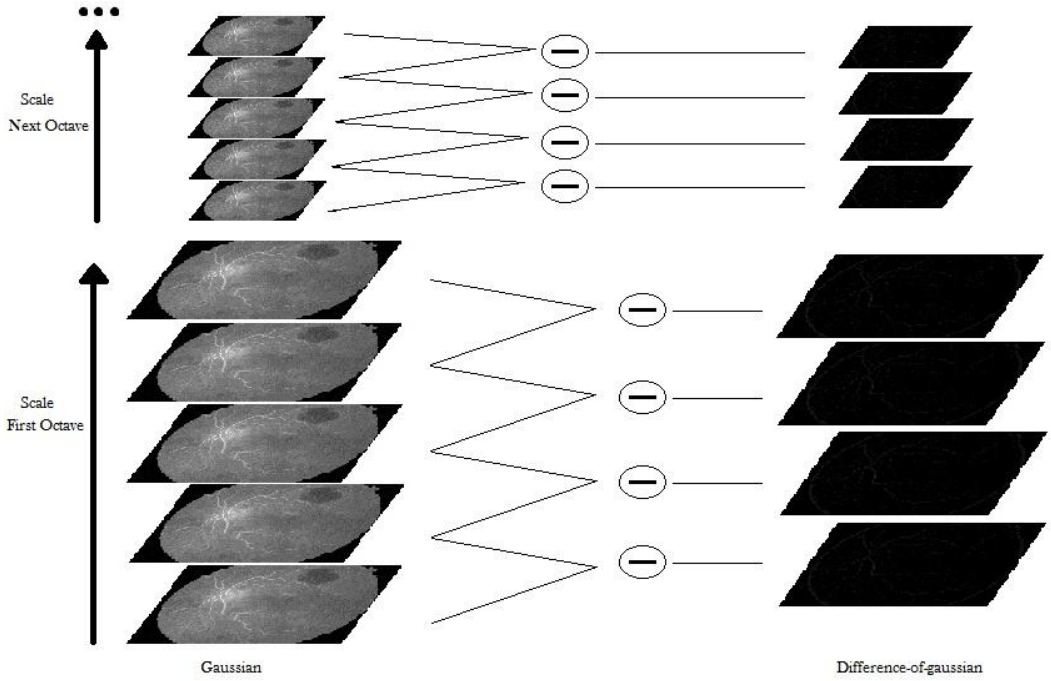


Figure 26 – Schematic representation of the DOG pyramid (scale-space function), based on figure 1 of [22].

Maxima and minima of this scale-space function are determined by comparing each pixel in the pyramid to its neighbours, figure 27. First, a pixel is compared to its 8 neighbours at the same level of the pyramid. If it is a maxima or minima at this level, then it is compared to the 9 neighbours of the level below. If it is still a maxima or minima it is compared to the above level pixels. Since most pixels will be eliminated within a few comparisons, the cost of this detection is small and much lower than that of building the pyramid [25].

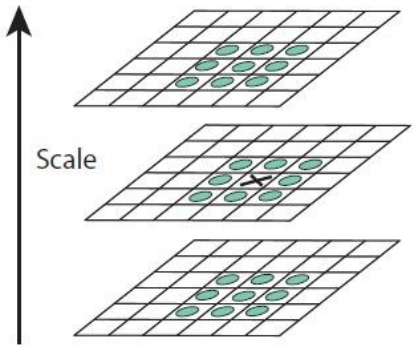


Figure 27 – Maxima and minima detection by comparing the pixel (marked with X) to its neighbours [22].

Not all the maxima and minima locations stand as possible keypoints. In order to achieve efficiency and stability, it is needed to reject points

with low contrast and poorly localized along an edge. Assuming that the image pixel values are in range [0,1], the extrema with absolute value less than 0,03 are discarded [22]. However, as the implementation was taking place, the results showed that in images with low contrast the algorithm find too few locations of possible keypoints, which compromises the efficiency of the matching method and so we tried an adaptive approach. The contrast threshold starts at 0,03 and the extrema are detected. If less than 150 points are detected, the threshold decreases and the search cycle is repeated. The search is performed while the threshold is higher than zero and if less than 150 points were detected.

For stability, it is not sufficient to reject points with low contrast. A poorly defined peak in the DOG function will have a large principal curvature across the edge but small one in the perpendicular direction. The principal curvature can be computed from a 2x2 Hessian matrix at the location and scale of the keypoint [22]:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The derivatives are estimated by taking differences of neighbouring sample points and the principal curvatures are determined by the trace and determinant of the Matrix *H*. Lowe [22] proposes the elimination of keypoints that have the curvature ratio higher than the value r ($r=10$) according to the following equation:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

Where,

$$Tr(H) = D_{xx} + D_{yy}$$
$$Det(H) = D_{xx}D_{yy} - \left(D_{xy}\right)^2$$

Furthermore, in the unlikely event that the determinant is negative, the curvatures have different signs so the point is also discarded as not being an extremum.

At this moment, the set of keypoints is counted and, as said above, if there are less than 150 keypoints the searching process is repeated decreasing the contrast threshold. It is important to notice that the accuracy of the transformation process is highly dependent of the success of the matching method, and for that a great number of keypoints is needed.

3.2. Orientation

One of the strengths of the SIFT approach is the rotation invariance of each keypoint, i.e. by assigning an orientation based on local image properties, the keypoint descriptor (explained later) can be represented relative to this orientation and therefore achieve invariance to image rotation.
In order to keep the keypoints scale-invariant, the Gaussian smoothed image, L, with the same scale of the current keypoint is selected. Then the gradient magnitude, $m(x,y)$, and orientation, $\theta(x,y)$, of each pixel of the current image is determined by:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)$$

As these gradients are computed to all the pixels, an orientation histogram is formed around the current keypoint. The histogram associated to the keypoint has 36 bins, where each represents 10 degrees, covering the 360 degree range of orientations. A sample weighted by its gradient magnitude and by a Gaussian-weighted circular window with σ equal to 1.5 times of the current scale is added to the histogram. This way, peaks in the orientation histogram correspond to the dominant directions of local gradients. The closest histogram values that are above and below of a peak are used to fit a parabola in order to interpolate the peak position for better accuracy. Furthermore, not only the highest peak is considered, but any other local peak that is within 80% of the highest one. In those cases, a new keypoint is created that has the same location and scale of the current keypoint, but a different orientation. This means that this specific location has multiple orientations, although only about 15% of points are assigned multiple orientations, which contribute significantly to the stability of the matching [22]. Figure 28 shows the detected keypoints with their gradient magnitude and orientation.

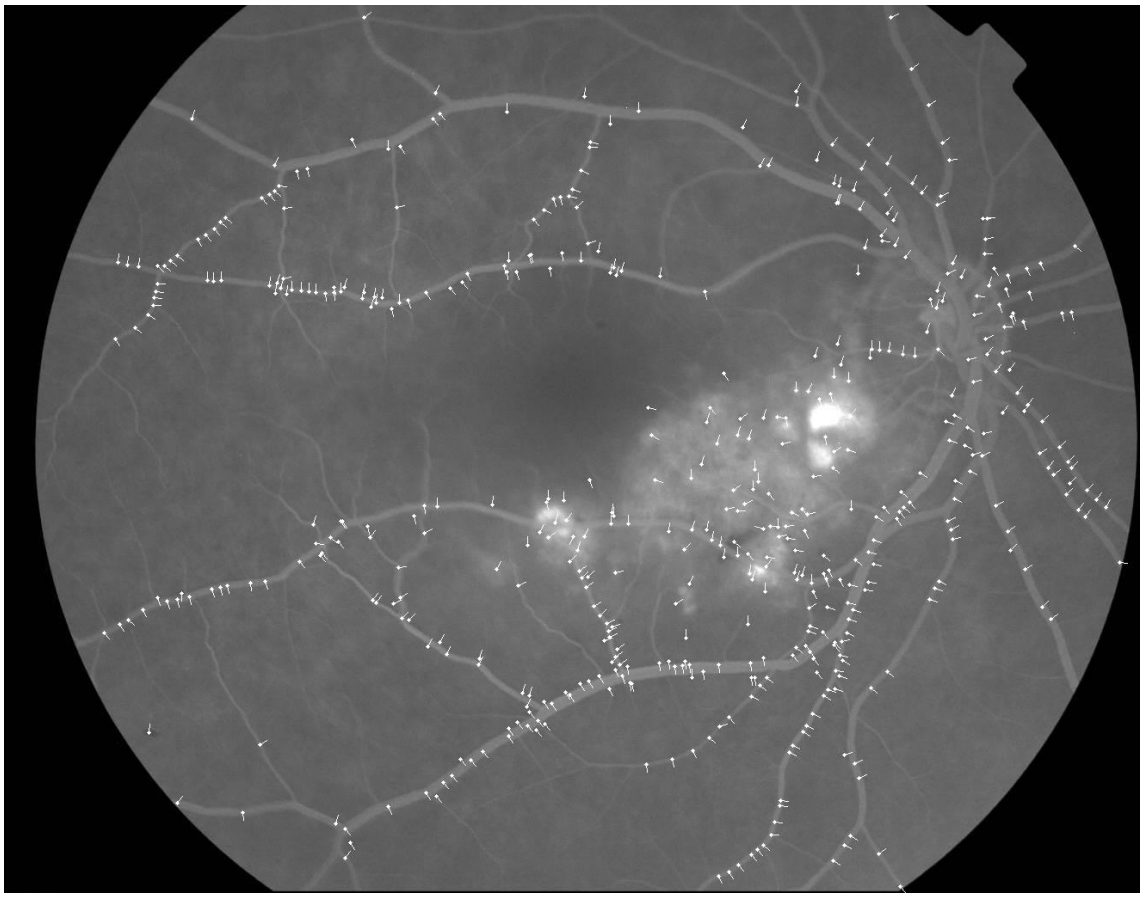


Figure 28 – Detected keypoints with the assigned gradient magnitude and orientation.

Notice that, as it is visible in the above picture, the detected points are mainly placed on top of the blood vessels and nerves, which means that they are more robust and reliable keypoints than the ones located in textureless areas.

3.3. Descriptor

At this stage, we have a set of keypoints defined by the coordinates (x,y) onto the original size image, the scale (where they were detected) and the local orientation and magnitude assigned in the latter method. The next step is to assign a local descriptor to each keypoint. This may be the most relevant part of the SIFT algorithm, as the descriptor is highly distinctive and will be the "central" feature for matching the keypoints.
A keypoint descriptor is a histogram of the gradient orientations of the pixels within a 16x16 window centred in the keypoint. First, the gradient magnitudes and orientations of those pixels are sampled, using the scale of the keypoints to select the level of the Gaussian smoothed image, in order to keep the scale invariance. On the other hand, the

orientation invariance can be achieved rotating the sampled gradient orientations relative to the keypoint orientation. Figure 29 illustrates the computation of the descriptor, although this method computes the descriptor from a 16x16 sample array the image only represents an 8x8 window.

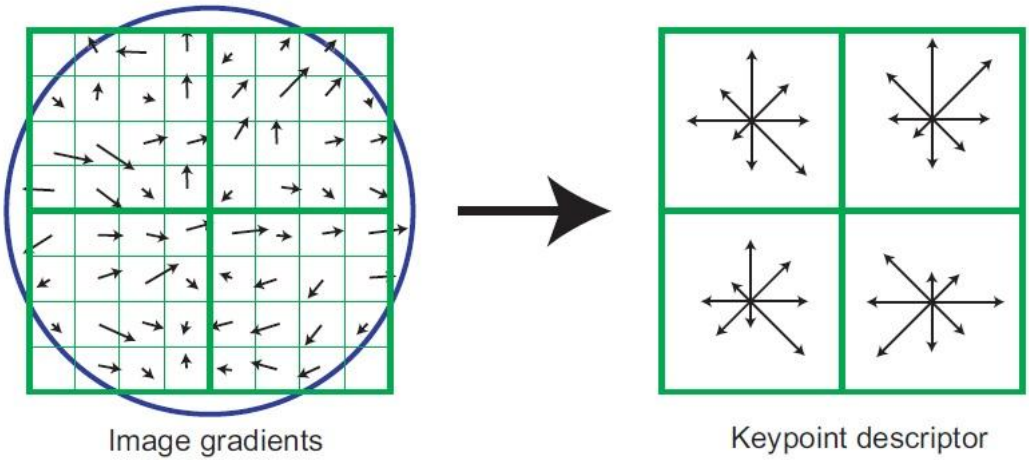


Figure 29 – SIFT descriptor representation [22].

As we can see on the left, each arrow represents the orientation of the gradient at that location and the size of the arrow is proportional to its gradient magnitude. The blue circle represents a Gaussian weighting function that avoids sudden changes in the descriptor with small changes in the position of the window, and gives less emphasis to gradients that are far from the centre of the descriptor. This Gaussian function must have σ equal to one half the width of the descriptor window (σ=8, as we use a 16x16 window) and is used to assign a weight to the magnitude of each sample point. On the right side of figure 29 is represented the descriptor histogram, where each square is the result of the combination of the gradient orientations of a 4x4 region on the left window. Each arrow covers a range of 45 degrees, so 8 arrows are needed to complete the 360 degree, as it is visible in each square. Therefore, to build this 8 bins histogram, the sum of the weighted magnitudes of the samples that are within the same orientation range is performed, i.e. all the magnitudes of the samples with an orientation within [0,45] range are summed to form that specific bin entry, and the length of the arrow correspond to the resultant magnitude. Notice that these orientations are relative to the current keypoint orientation. In the end we will have a 4x4x8 array histogram, which corresponds to a 128 bin histogram. To ease the matching process, this array is rearranged into a vector of 1x128 [22].

In order to reduce the effects of illumination change (contrast and brightness) and the influence of large gradient magnitudes, two normalizations were performed. To the former, a normalization of the descriptor vector to the unit length is sufficient. The contrast changes can be cancelled by the normalization, as they are result of a multiplication of a constant value to each gradient. The brightness change, in turn, is a constant added to each pixel, which will not affect the gradient values as they are computed from pixel differences. To reduce the influence of the latter, it is applied a threshold to the unit feature vector, so each value be no larger than 0,2, and then normalize the vector again to the unit length. This means that the matching of the distribution of orientations has greater emphasis than the magnitudes for large gradients [22].

3.4. Matching

The whole point of having such a distinctive and invariant set of features is the possibility of having a point-to-point match between two images of the same scene. The performances of the matching method, as well as the false positive elimination method, are determinant to the success of the registration process.

At this moment we have two images (a reference and a sample image), a set of features that define each keypoint, and we want to have a set of reliable and robust matching pairs in order to align both images through registration. The best candidate match for each keypoint, in the reference image, is found by identifying its nearest neighbour in the set of keypoints of the sample image. The nearest neighbour is defined as the keypoint with minimum Euclidean distance for the descriptor according to the following equation:

$$d\left(D_i^r, D_j^s\right) = \sqrt{\sum_{k=1}^{128}\left(D_i^r[k] - D_j^s[k]\right)^2}$$

with i=1,…,n_1 and j=1,…,n_2, where n_1 and n_2 are the number of keypoints in the reference and sample images, respectively. D_i^r is the descriptor for the keypoint i of the reference image and D_j^s the descriptor of keypoint j of the sample image.

In order to discard features that do not have any good match, Lowe [22] proposes a comparison between the distances of the closest neighbour

to that of the second-closest neighbour. So all matches in which the distance ratio $\left(\frac{d_{closest\ neighbour}}{d_{2nd-closest\ neighbour}}\right)$ is greater than 0,8 are rejected.
With the first matching results, it was identified the issue that keypoints near the image mask (see appendix A) were almost always mismatched. In order to minimize this problem, code was implemented to determine the image mask and then reject the matches that belong to the mask or are up to 10 pixels radius near the mask. Furthermore, to obtain the matched pairs of keypoints, a bilateral matching method was performed [26]. In the first place, a set of unilateral matches is determined where it is searched a match to each keypoint of the reference image, through the aforementioned method. However, it does not exclude the case where two keypoints in the reference image are matched to the same keypoint in the sample image. So we create a second set of unilateral matches, but this time it is searched a match to each keypoint of the sample image. Then, we only need to select the same exact matches on both sets, creating the bilateral match set. Although we exclude some mismatches with the bilateral matching method, it is not guaranteed that all matches are correct and further false positive elimination methods need to be applied.

3.4.1. False Positive Elimination

The false positive elimination method is performed in two parts: first the orientation differences of the keypoints and then the distance and angle between the matched points. The former is based on the assumption that the difference between the orientations of each match is approximately constant [26]. Thus, we determine the median of the differences between orientations and compare this value to the difference of each match. If a match differs to the median more than 10 degree, the match is discarded.
The second part of the method consists in the determination of the distance and angle between each match. For a better understanding, figure 30 is a schematic representation of the mentioned distance and angle. Although we cannot assume that the distance and angle between each match are constant for all the matches, due to the possibility of non-rigid deformations in this type of images, we can exclude some mismatched keypoints that are significantly far from the mean distance and angle. In the first place, we determine the distance and angle between the points of a match according to the following equations:

$$D_k = \sqrt{{d_x}^2 + {d_y}^2}$$

$$\theta_k = tan^{-1}\left(\frac{d_y}{d_x}\right)$$

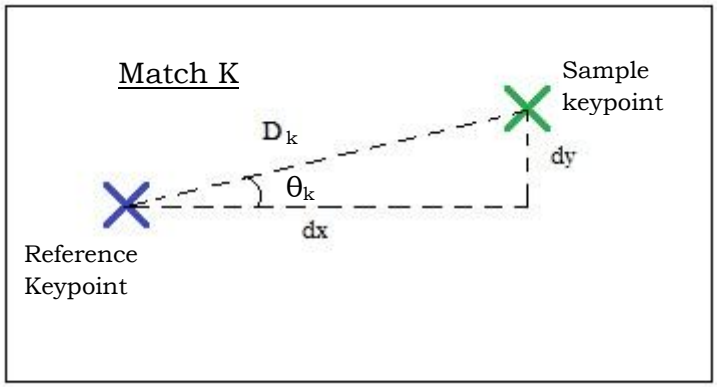


Figure 30 – Representation scheme of the distance and angle between the points of a match

Once the mean distance and angle are computed, we discard the matches that differ from the mean more than a certain threshold value. These thresholds were experimentally determined, using a set of ten image pairs, where we have chosen those with the higher specificity and sensibility. As the polynomial transformation is very unstable by itself, our goal was to successfully eliminate all the wrong matches, even if some well-matched keypoints were excluded. In other words, the method needed to have a great specificity and the better possible sensibility. The obtained results are shown in the table on appendix B.
In order to see the result of the false positive elimination and to prove the need of the second part of the method, in figure 31 – a) it's represented the matches after the orientation difference elimination. The red lines show the false positives and the white lines represent the true matches. As we can see in figure 31 – b) after applying the second part of the method, the false matches were successfully eliminated.

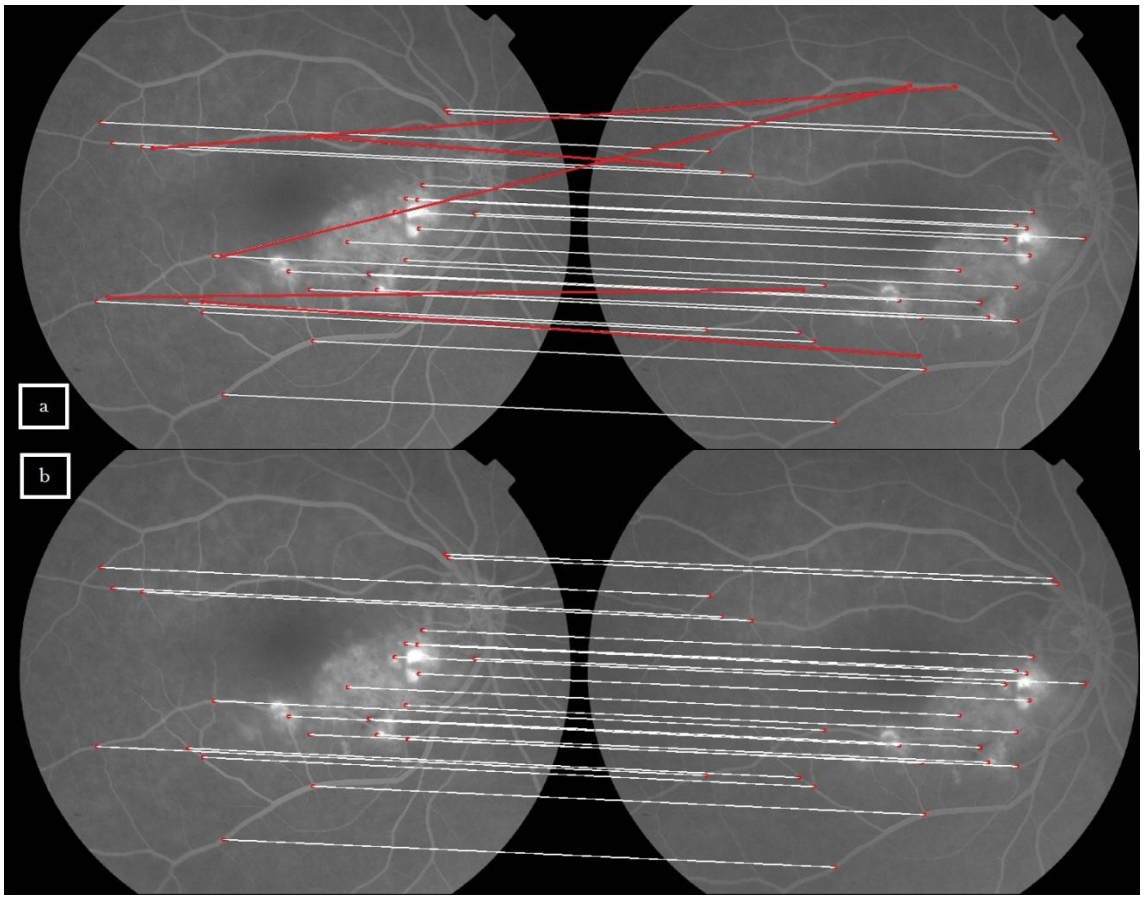


Figure 31 – Matched keypoints representation. A) after false positive elimination by orientation difference. B) after false positive elimination by distance and angle between points.

3.5. Registration algorithm

The previous sections of this third chapter described the methods to find the matched keypoints needed to solve the polynomial transformation model and therefore perform the alignment. Unlike the manual registration methods, we have more than 6 registration points for each image, so we used a squared error minimization method to obtain the best-fitting transformation coefficients.
Recalling the polynomial transformation model, it is given by the following equation:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 & \theta_{10} & \theta_{11} & \theta_{12} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \\ 1 \end{bmatrix}$$

As we wish to solve for the transformation parameters, θ, the above equation can be rewritten as:

$$\begin{bmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2y_2 & x_2 & y_2 & 1 \\ & & \cdots & & & \\ x_n^2 & y_n^2 & x_ny_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \cdots \\ u_n \end{bmatrix}$$

Notice that, for an easy understanding, only the parameters for xx transformation are represented. This linear system can be presented as

$$Ax = b$$

where the solution for the parameters, **x**, can be determined by solving the correspondent normal equations [25]:

$$x = [A^T A]^{-1} A^T b$$

Although we can get a unique solution for the transformation coefficients, the polynomial model does not perform well with so many matched keypoints. As mentioned before, slight differences between a match keypoints contribute to the unpredictable behaviour of the model and as more matches are used, the probability of introducing those errors increase. As the first registration results started to show up, we verified that most of the image pairs were not properly aligned.
Therefore, we decided to try a different approach to solve the linear system based on the k-means clustering and the squared error to determine the best-fitting transformation coefficients.

Pseudo-code:

- ✓ While there are more than 6 matches:
  - Run the K-means algorithm to find 6 clusters;
  - Compute the transformation matrix from the 6 matches;
  - Determine the error for each keypoint;
  - Hold the squared error for the current transformation matrix;
  - Delete the match with higher error;
- ✓ Perform the alignment of the sample image with the transformation matrix with the lower square error.

The K-means algorithm distributes a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster. First, the algorithm randomly selects k of the objects, each of which initially represents a cluster center. The remaining objects are assigned to the cluster to which is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster and iterates again to assign the objects to the new cluster. The process is stopped when there are no more changes in the clusters [27]. Now each match belongs to a cluster but we need to select a match within each cluster. As said, the polynomial model is unstable when the points to determine the transformation coefficients are too close to each other. So we decide to determine the mean coordinate of all matches and select the match of each cluster that is farthest from the mean. With the six matching keypoints we compute the transformation matrix and hold the correspondent square error, which is calculated according to the following equation [28]:

$$E = \sum_{i=1}^{N} (y_i - ax_i)^2$$

where a is the transformation matrix, x_i is the coordinate of match i of the sample image, y_i is the coordinate of match i of the reference image and N represents the total number of matches. Then, we delete the match with the higher error and iterate again. This way we will have several combinations of six pairs of matches, and will be able to choose the transformation coefficients that minimizes the square error.

With the transformation matrix determined, we must apply it to align the images. The whole process is exactly the same of the manual registration described in section *2.1.1* of the second chapter, so it will not be presented again in here.

To conclude the automatic registration chapter, figure 32 is the result of the alignment of a pair of images of the used dataset. As explained in the chapter 2, the pseudo-colour has as its main purpose the ease of the accuracy evaluation of the alignment, where the reference and aligned sample images correspond to the green and blue channels respectively. The automated registration algorithm was tested with a dataset of 10 pairs of images, where 3 of them are angiographies and the remaining 7 are retinographies. However, as mentioned before, within each pair the images correspond to the same image modality. The aligned images are represented in appendix C, where we can see

that 5 of the pairs were successfully aligned, where 3 of those are angiographies. Since the algorithm will be a tool to assist the physicians, it makes sense to know the total time that the process takes to return the aligned image. The computational cost varies according to the images size and the extent of the keypoints detected, which depends on the image modality and quality. So, the mean of the total time for the algorithm return the aligned image is close to 2 minutes and 30 seconds for images with a size of 1300x1000.

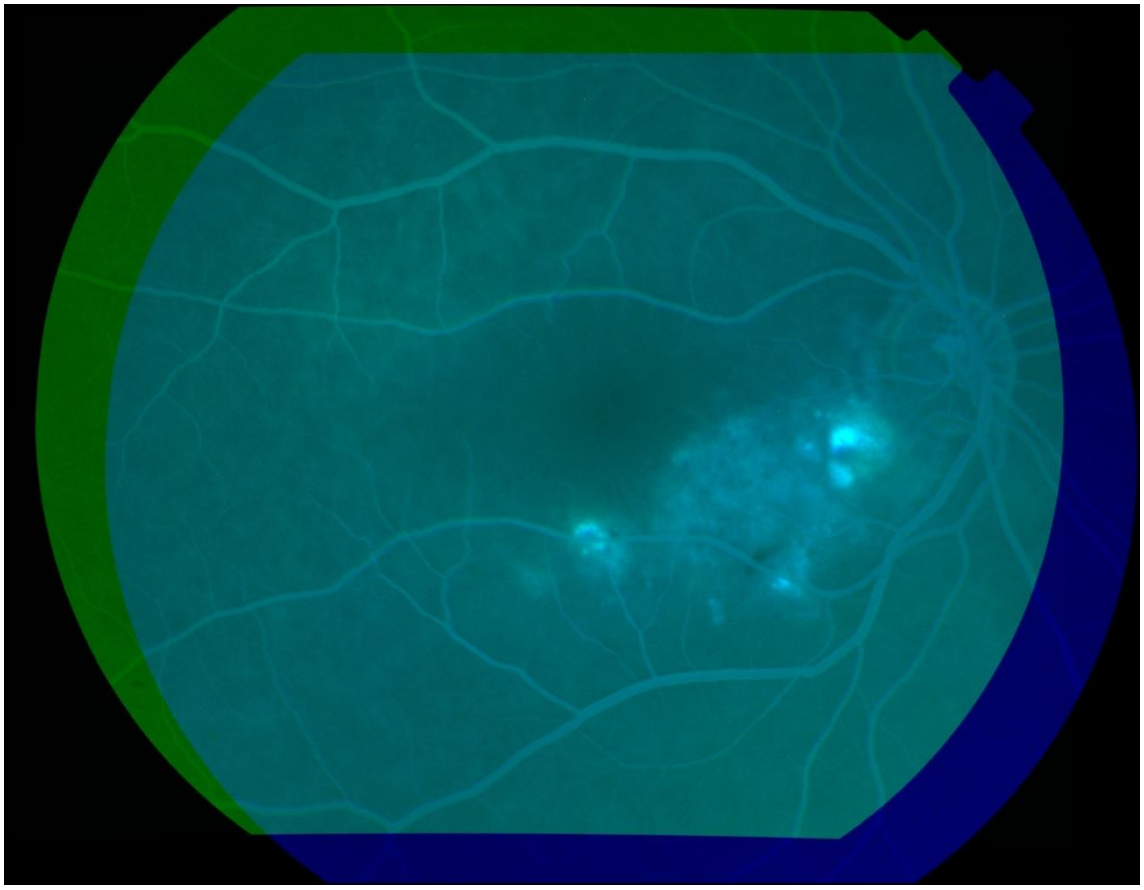


Figure 32 – Alignment result of the automatic registration.

Chapter 4

Conclusions

Looking back to the developed work, some conclusions can be taken. In the first place, the manual registration GUI for the alignment of ocular fundus images already brought results. A North American start-up enterprise approached BW in order to get software to align multiple images of the ocular fundus and overlay them into a single image with different transparency levels. Our solution was then adapted to run as a stand-alone application and the GUI was also modified to answer their demands.
On the other hand, the perimetry registration GUI was slightly "forgotten". After concluding the GUI, we sent the results to the ophthalmologists that were present in the brainstorming meeting for a first evaluation and to know how we could improve the algorithm, however we did not get feedback.
Last but not least, the automated registration implementation was partially successful, once five out of ten good alignments are not the expected results. Though, this algorithm has two limitations that could compromise its integration in the OphthalSuite: computation time; the polynomial transformation unpredictable behaviour. Taking into account that the algorithm is to be used by the ophthalmologists during consults, the ideal is to have the aligned images instantly. However, as said before the automatic method takes an average of 2 minutes and 30 seconds to return the aligned image. This may not seem a lot, but the manual GUI takes less than a minute (depending on the users experience to use the GUI). This is only a problem if physicians think so and we are still waiting for their feedback. The latter limitation is highly dependent on the images quality (pixel definition and contrast). Almost all of the references cited, mention that the affine model is used because of the unpredictability of the polynomial model. However, as

explained earlier, the affine transformation is not sufficient to correctly align the retinal images.

So, as future work, the translation problem with the *cvRemap()* function needs to be better solved. The implemented solution of adding the black bands to the images was a last resource, once it is not an efficient resolution. To improve the number of matches detected, a pre-processing method to enhance image contrast could help to achieve better results in the alignment. However, the multimodality between images is a problem that hardly can be overtake by the SIFT method. As said, the SIFT keypoints are scale-space and orientation invariant but depend on the local features. When we try to match keypoints from images that are from different modalities, the result is bad because of the local information that is quite different in both images. In this case, a different approach should be a better way.

Appendix A

Mask Determination

The mask determination algorithm is divided in four stages represented in figure 33. Notice that the mask is the black shadow around the image created by some image acquisition systems. Those pixels are not totally black but have an intensity value up to 15 (in the interval [0,255]). So the mask determination method is nothing more than mark the pixels with value lower than 15 as belonging to the mask. The first stage analyses each row of the image from left to right, marking the pixels as mask until a pixel with value higher than 15 is found. When this happens, the method moves to the next row. When all lines are analysed, the second stage begins. In this stage the rows are analysed form right to left. The third and fourth stages compute the columns from top to bottom and from bottom to top, respectively. The result is a binary map that separates mask pixels from the actual ocular fundus image [12].

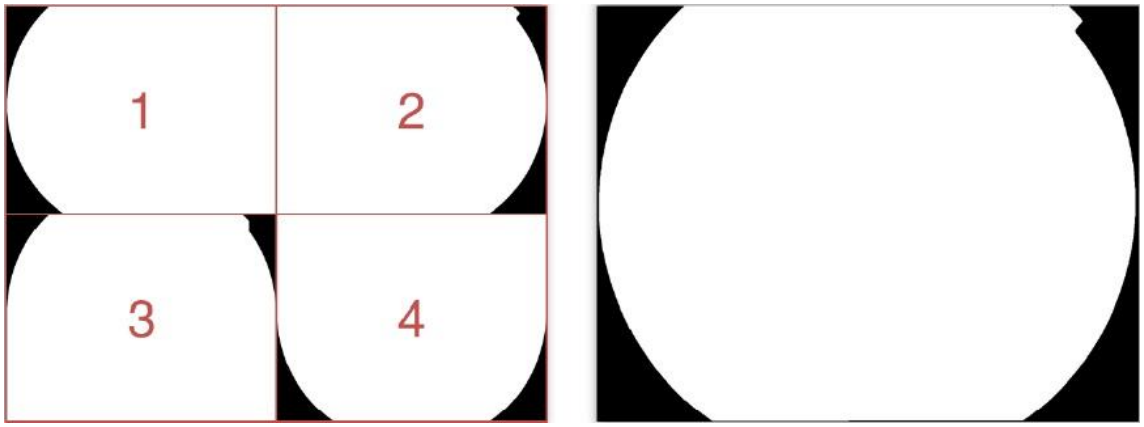


Figure 33 – Mask determination illustration. Left: four steps of the algorithm. Right: mask resulting from the intersection of the four steps.

Appendix B

False Positive Elimination

		D-m/2 & θ-10º	D-2m/3 & θ-10º	D-3m/4 & θ-10º	D-4m/5 & θ-10º	D-m & θ-10º	D-0.9*m & θ-10º
Pair 1	Initial Matches	65	65	65	65	65	65
	TN	8	8	8	8	7	8
	FN	0	0	0	0	0	0
	TP	57	57	57	57	57	57
	FP	0	0	0	0	1	0
	Final Matches	57	57	57	57	58	57
	Sensitivity	1	1	1	1	1	1
	Specificity	1	1	1	1	0,875	1
Pair 2	Initial Matches	21	21	21	21	21	21
	TN	2	2	2	2	2	2
	FN	0	0	0	0	0	0
	TP	19	19	19	19	19	19
	FP	0	0	0	0	0	0
	Final Matches	19	19	19	19	19	19
	Sensitivity	1	1	1	1	1	1
	Specificity	1	1	1	1	1	1

Pair 3	Initial Matches	11	11	11	11	11	11
	TN	4	4	4	4	4	4
	FN	3	3	3	3	3	3
	TP	4	4	4	4	4	4
	FP	0	0	0	0	0	0
	Final Matches	4	4	4	4	4	4
	Sensitivity	0,571428571	0,571428571	0,571428571	0,571428571	0,571428571	0,571428571
	Specificity	1	1	1	1	1	1
Pair 4	Initial Matches	69	69	69	69	69	69
	TN	3	3	3	3	3	3
	FN	0	0	0	0	0	0
	TP	66	66	66	66	66	66
	FP	0	0	0	0	0	0
	Final Matches	66	66	66	66	66	66
	Sensitivity	1	1	1	1	1	1
	Specificity	1	1	1	1	1	1
Pair 5	Initial Matches	13	13	13	13	13	13
	TN	2	2	2	2	1	1
	FN	11	10	6	6	6	6
	TP	0	1	5	5	5	5
	FP	0	0	0	0	1	1
	Final Matches	0	1	5	6	6	6
	Sensitivity	0	0,090909091	0,454545455	0,454545455	0,454545455	0,454545455
	Specificity	1	1	1	1	0,5	0,5
Pair 6	Initial Matches	35	35	35	35	35	35
	TN	2	2	2	2	2	2

	FN	11	11	11	11	11	11
	TP	22	22	22	22	22	22
	FP	0	0	0	0	0	0
	Final Matches	22	22	22	22	22	22
	Sensitivity	0,666666667	0,666666667	0,666666667	0,666666667	0,666666667	0,666666667
	Specificity	1	1	1	1	1	1
Pair 7	Initial Matches	24	24	24	24	24	24
	TN	1	1	1	1	1	1
	FN	17	17	17	17	17	17
	TP	6	6	6	6	6	6
	FP	0	0	0	0	0	0
	Final Matches	6	6	6	6	6	6
	Sensitivity	0,260869565	0,260869565	0,260869565	0,260869565	0,260869565	0,260869565
	Specificity	1	1	1	1	1	1
Pair 8	Initial Matches	49	49	49	49	49	49
	TN	4	4	4	4	4	4
	FN	2	2	2	2	2	2
	TP	43	43	43	43	43	43
	FP	0	0	0	0	0	0
	Final Matches	43	43	43	43	43	43
	Sensitivity	0,955555556	0,955555556	0,955555556	0,955555556	0,955555556	0,955555556
	Specificity	1	1	1	1	1	1
Pair 9	Initial Matches	30	30	30	30	30	30
	TN	2	2	2	2	2	2
	FN	16	12	12	12	12	12
	TP	12	16	16	16	16	16

	FP	0	0	0	0	0	0
	Final Matches	12	16	16	16	16	16
	Sensitivity	0,428571429	0,571428571	0,571428571	0,571428571	0,571428571	0,571428571
	Specificity	1	1	1	1	1	1
Pair 10	Initial Matches	29	29	29	29	29	29
	TN	5	5	5	5	5	5
	FN	0	0	0	0	0	0
	TP	24	24	24	24	24	24
	FP	0	0	0	0	0	0
	Final Matches	24	24	24	24	24	24
	Sensitivity	1	1	1	1	1	1
	Specificity	1	1	1	1	1	1

Sensit_mean	0,688309179	0,711685802	0,748049438	0,748049438	0,748049438	0,748049438
specif_mean	1	1	1	1	0,9375	0,95

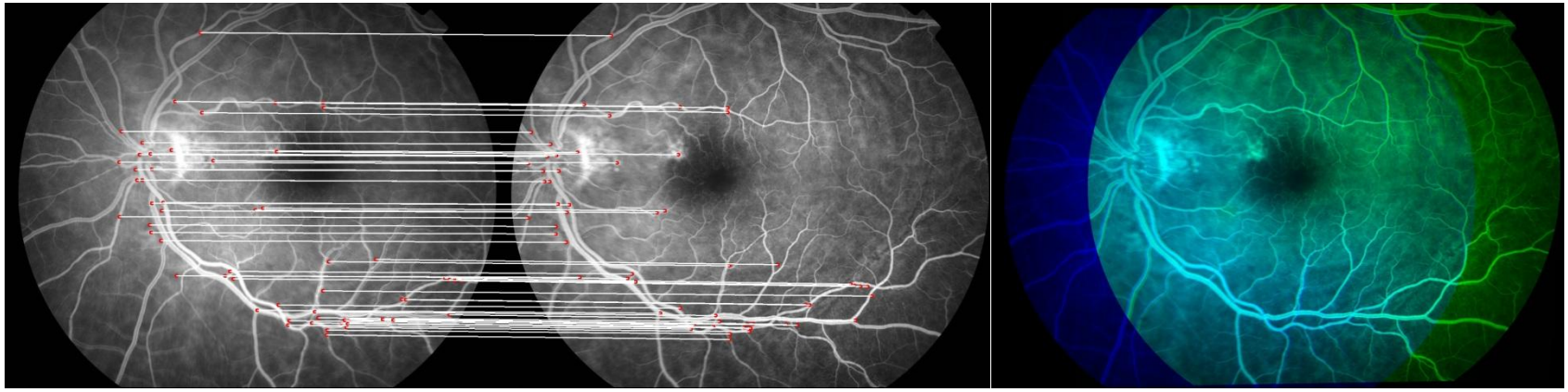
The parameter θ is the threshold value for the angle between the points of a match. Although it is not showed in this table, we tried for θ=10°, θ=15° and θ=20° and the better results were obtained with the θ=10° threshold.
The parameter D is the threshold value for the distance between the points of a match, where the 'm' denotes the mean distance calculated. As we can see the best results were obtained with a distance threshold equal to $\frac{3m}{4}$.

Appendix C

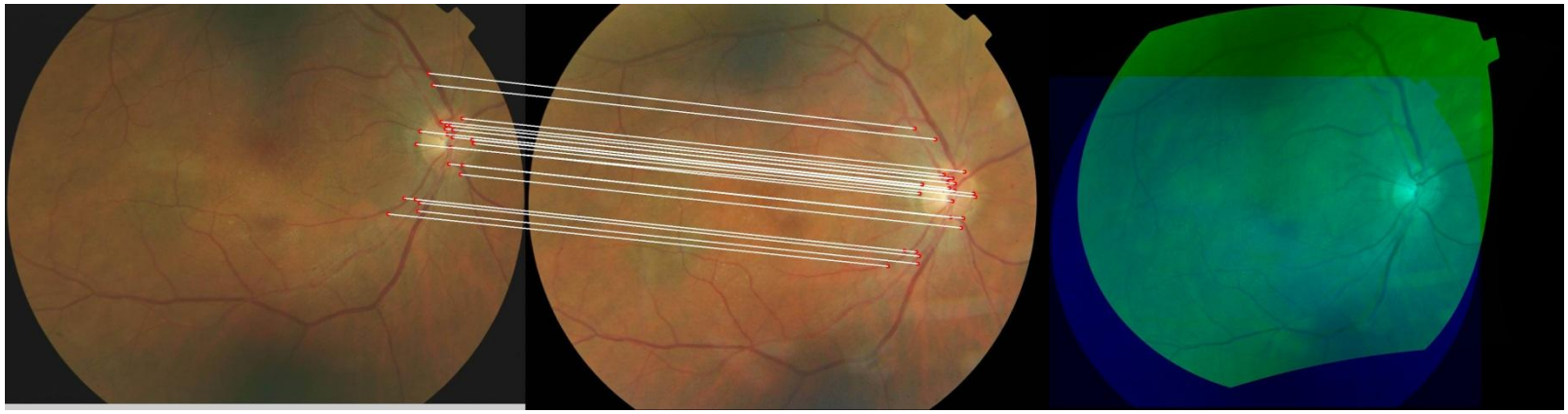
Automatic Registration Image Results

The results of the automatic registration are presented, where on the left it's represented the matches and on the right the overlay of the reference and aligned sample images. Notice that this dataset was gathered within the BlueWorks archives.

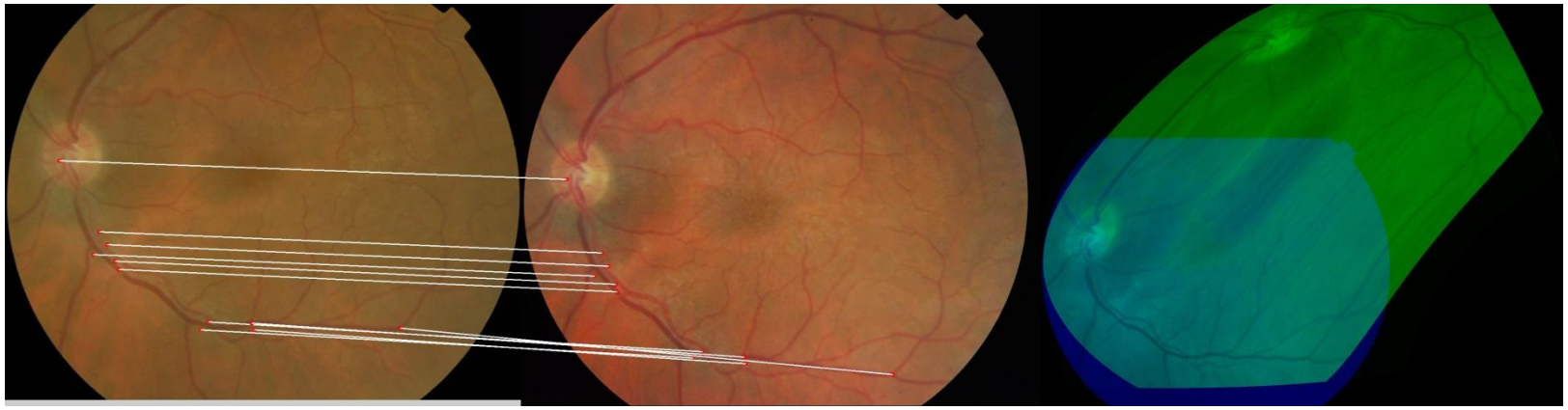
1st Pair



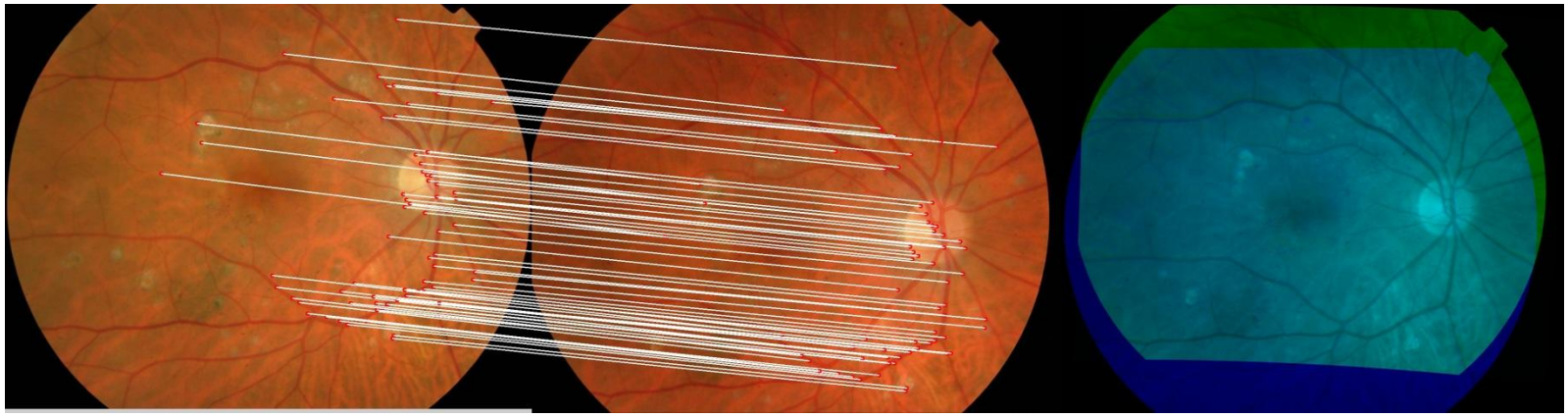
2nd Pair



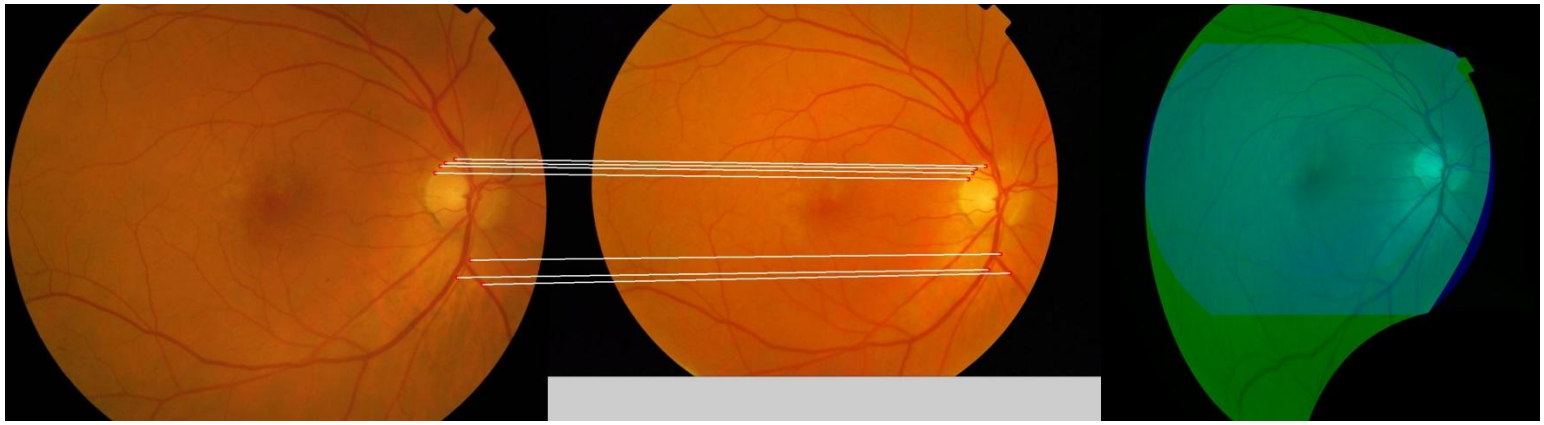
3rd Pair



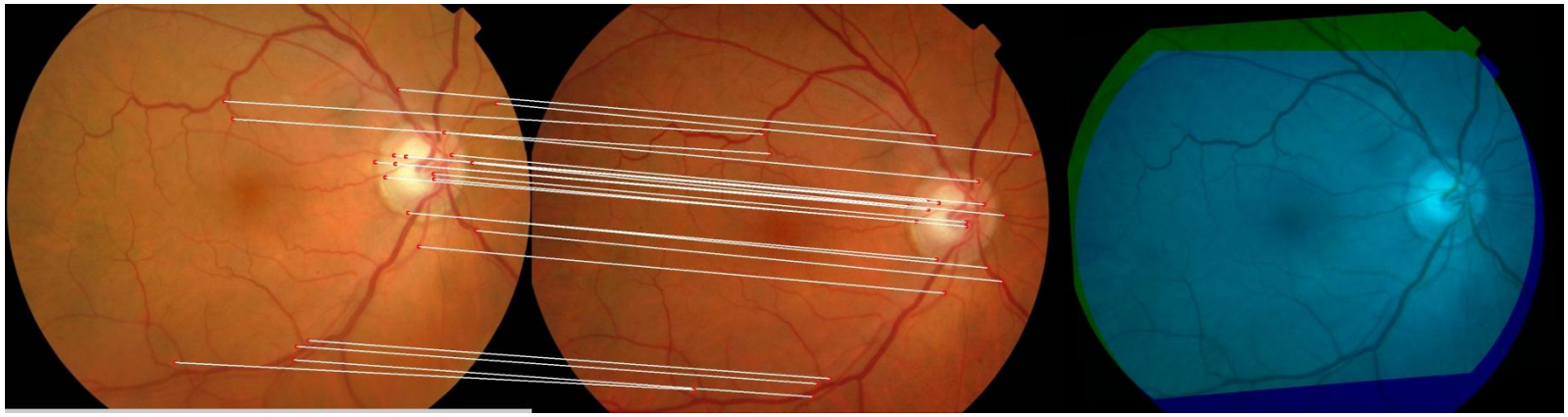
4th Pair



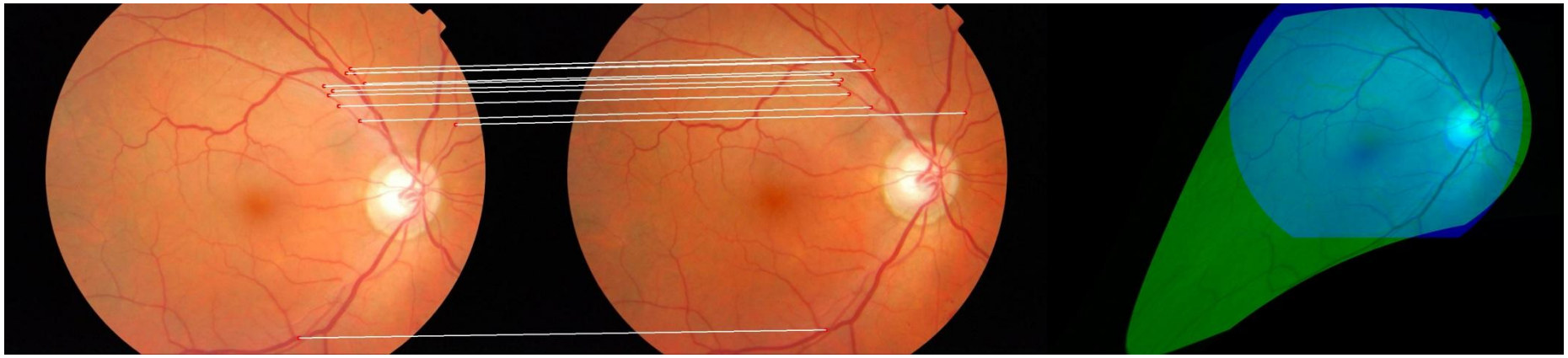
5th Pair



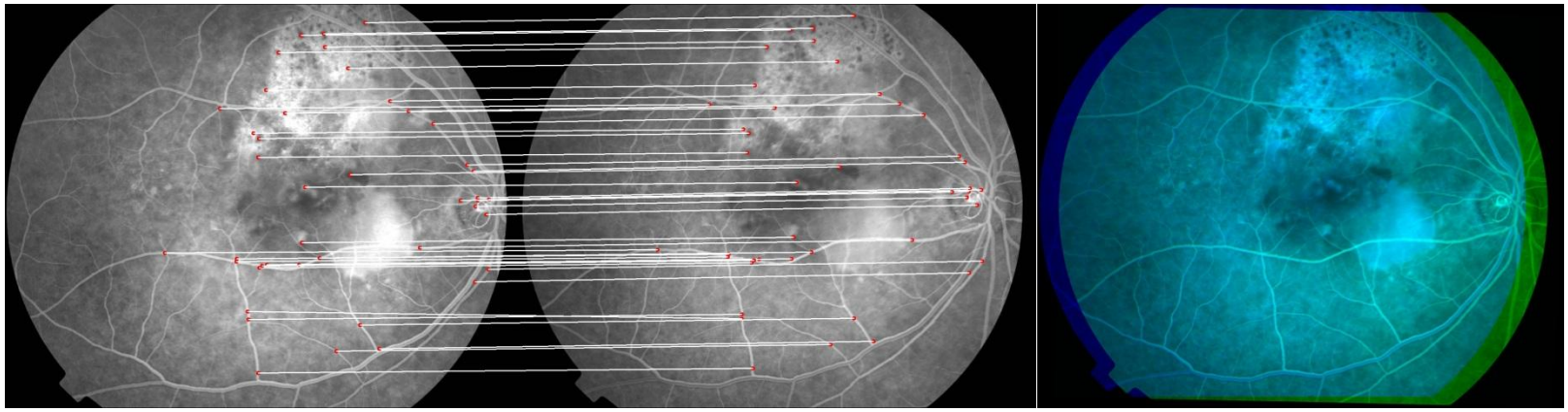
6th Pair



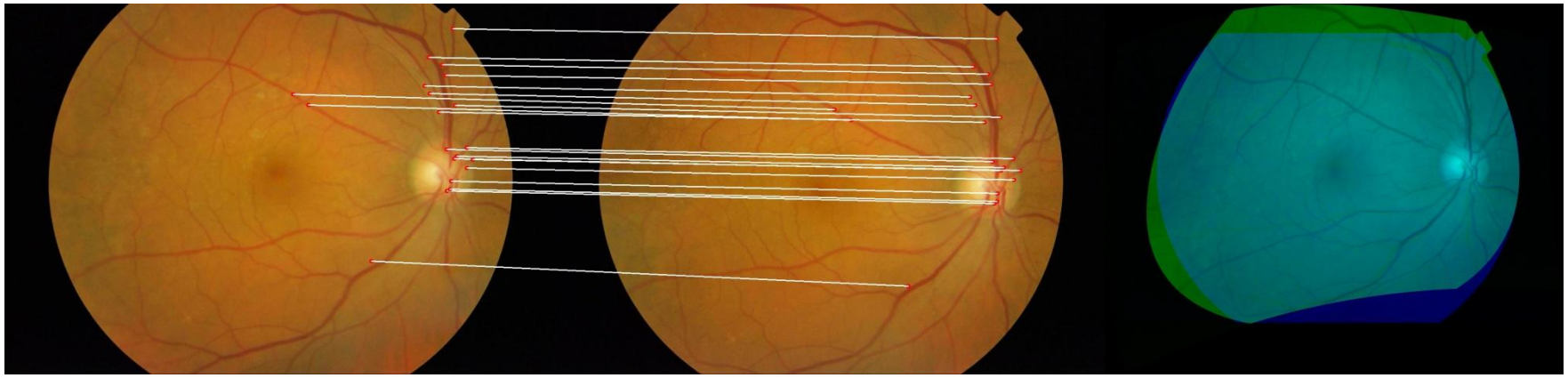
7th Pair



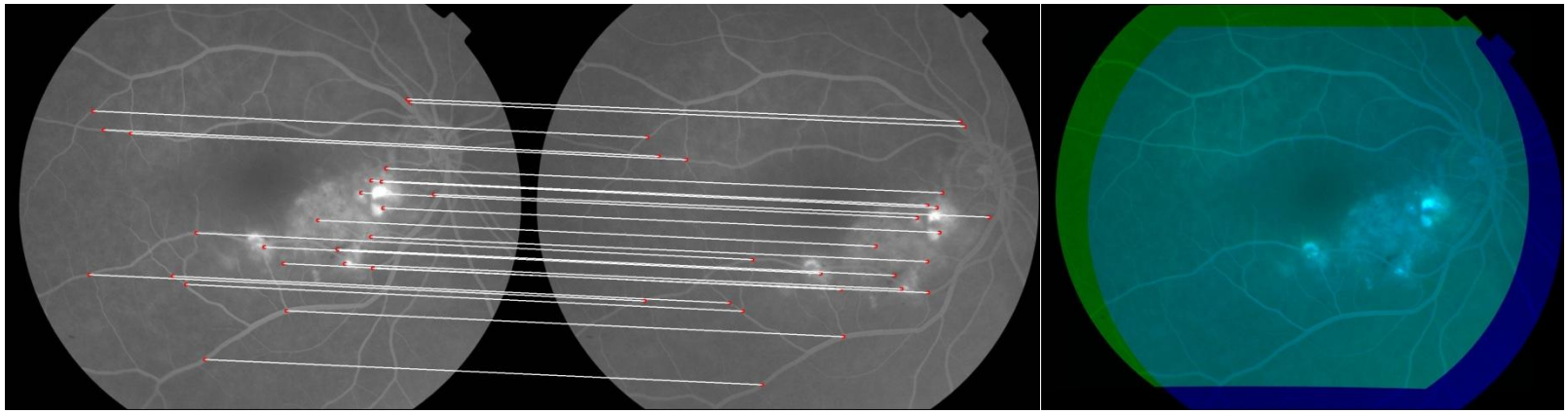
8th Pair



9th Pair



10th Pair



References

[1] R. R. Seeley and T. D. T. P. Stephens, Anatomia e Fisiologia, 6ª edição, McGraw - Hill Companies, Inc, 2003.

[2] L. C. Junqueira and J. Carneiro, Basic Histology: text and atlas, 11th edition, McGraw-Hill Medical, 2005.

[3] C. A. Bradford, Oftalmología básica, 1ª edición, El Manual Moderno, 2005.

[4] G. K. Lang, Ophthalmology: a short textbook, New York: Thieme Stuttgart, 2000.

[5] "Eye anatomy, Ocular anatomy, Vision conditions & problems," [Online]. Available: http://www.mastereyeassociates.com/eye-anatomy-eye-problems/. [Accessed March 2012].

[6] "Fundus Photography overview - Ophthalmic Photographers' Society," [Online]. Available: http://www.opsweb.org/?page=fundusphotography. [Accessed June 2012].

[7] "Angiography - Ophthalmic Photographers' Society," [Online]. Available: http://www.opsweb.org/?page=Angiography. [Accessed June 2012].

[8] T. J. Bennett, "Fluorescein Fundamentals - Ophthalmic Photographers' Society," [Online]. Available: http://www.opsweb.org/?page=FA. [Accessed June 2012].

[9] B. Fischer and J. Modersitzki, "Ill-posed medicine - an introduction to image registration," *IOP Publishing, Inverse Problems, 24,* 2008.

[10] A. Can, C. V. Stewart, B. Roysam and H. L. Tanenbaum, "A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina," *IEEE Transactions on Pattern Analysis and Machine Inteligence, vol.24, no.3,* pp. 347-364, 2002.

[11] L. G. Brown, "A survey of image Registration Techniques," *ACM Computing Surveys,* vol. 24, no. 4, pp. 325-376, 1992.

[12] T. Chanwimaluang, G. Fan and S. R. Frasen, "Hybrid Retinal Image Registration," *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, no. 1, pp. 129-142, 2006.

[13] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing,* vol. 21, pp. 977-1000, 2003.

[14] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai and M. Xu, "Retinal Fundus Image Registration via Vascular Sturcture Graph Matching," *International Journal of Biomedical Imaging,* pp. 1-13, 2010.

[15] A. Bhuiyan, E. Lamoureux, B. Nath, K. Ramamohanarao and T. Y. Wong, "Retinal Image Matching Using Hierarchical Vascular Features," *Computacional Intelligence and Neuroscience,* pp. 1-7, 2011.

[16] L. Chen, Y. Xiang, Y. Chen and X. Zhang, "Retinal Image Registration Using Bifurcation Structures," *IEEE International Conference on Image Processing,* vol. 18, pp. 2169-2172, 2011.

[17] H. Gonçalves, J. A. Gonçalves and L. Corte-Real, "HAIRIS: A Method for Automatic Image Registration Through Histogram-Based Image Segmentation," *IEEE Transactions on Image Processing, vol. 20, no. 3,* pp. 776-789, Mach 2011.

[18] A. Bardera, M. Feixas, I. Boada and M. Sbert, "Image registration by compression," *Information Sciences,* vol. 180, pp. 1121-1133, 2010.

[19] H. Sarnel and Y. Senol, "Accurate and Robust image registration based on radial basis neural networks," *Neural Comput & Applic,* vol. 20, pp. 1255-1262, 2011.

[20] J. Yang, J. P. Williams, Y. Sun, R. S. Blum and C. Xu, "A Robust hybrid method for nonrigid image registration," *Pattern Recognition,* vol. 44, pp. 764-776, 2011.

[21] H. Gonçalves, L. Corte-Real and J. A. Gonçalves, "Automatic Image Registration Through Image Segmentation and SIFT," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 49, no. 7, pp. 2589-2600, 2011.

[22] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* pp. 1-28, 2004.

[23] C. A. Johnson, M. Wall and H. S. Thompson, "A History of Perimetry and Visual Field Testing," *Optometry and Vision Science,* vol. 88, no. 1, pp. E8-E15, 2011.

[24] S. Al-Abed, "Perimetry - Ophthalmology 101," 2011. [Online]. Available: https://sites.google.com/site/ophthalmology101/common-investigations/perimetry. [Accessed 20 June 2012].

[25] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *International Conference on Computer Vision, Corfu,* pp. 1-8, 1999.

[26] J. Chen, R. T. Smith, J. Tian and A. F. Laine, "A novel registration method for retinal images based on local features," *Conf Proc IEEE Eng Med Biol*

Soc., 2008.

[27] J. Han and K. Micheline, Data Mining Comcepts and Techniques, 2º ed., Morgan Kaufmann Publishers, 2006, pp. 402-403.

[28] S. J. Miller, *The Method of Least Squares,* Mathematics Department Brown University Providence, RI 02912.