Samuele Fiorini

Automatic generation of Retinal Fundus Image Phantoms: non-vascular regions

Tesi di Laurea Magistrale
Relatore: Prof. Alfredo Ruggeri
Correlatore: Prof. Emanuele Trucco

Università degli Studi di Padova
Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione
2013 - 2014

To my family.

I love deadlines. I love the whooshing noise they make as they go by.

Douglas Adams

Abstract

Retinal fundus imaging is a very helpful tool for the diagnosis of many diseases such as diabetes, glaucoma and cardiovascular conditions. The advent of digital fundus cameras has given the opportunity for computer vision techniques to be applied to the field of ophthalmology. Algorithm validation on ground truth images is a well documented and necessary practise. However, it is also well known that obtaining large quantities of annotated medical ground truth images is an expensive and laborious task. This study aims to generate novel realistic retinal fundus colour images, similar in characteristics to a given dataset, as well as the corresponding ground truth mask. A representative task could be, for example, the synthesis of a realistic retinal image with the corresponding vessel tree and optic nerve head binary map, measurement of vessel width in any position, macula localisation and so on. All these measurements are of crucial importance during the validation of retinal image analysis algorithms. In the synthesised retinal phantoms, textural anatomical features can be modified to simulate a wide range of parameters, as well as different populations. Both patch-based and model-based texture synthesis techniques have been developed and implemented. This work mainly focuses on the generation of non-vascular regions, and it is complemented by a parallel study which aims at generating structure and texture of the vessel network. The presented technique has been developed on the publicly available HRF database and results from the VAMPIRE software suite have also been used. The method is implemented in 64 bit MATLAB® 2012b.

Keywords: Retinal Fundus Imaging, Texture Synthesis, Medical Imaging Phantom, Ground Truth

Acknowledgements

I would like to express my deepest appreciation to all the components of the VAMPIRE team, Manuel, Lucia, Enrico, Roberto and Kris who, with their wise advice, have helped me in this journey and have introduced me to the research world. I am very grateful to my supervisor Prof. Manuel Trucco: thank you for all the time you spent helping me to solve my doubts.

A special thanks to my family, I will never repay you enough for all the support you always give me. To my loving and supporting girlfriend, Claudia: my heartfelt thanks. Thank you for being always by my side, even from one thousand miles, *a tié proprio la mejo d'tuti*. The next sincere thanks goes to Francesca, who has helped me with the proofreading of this thesis. I would also thank to all my friends, for the help and the encouragement they always give me. And last, but not least, to all the wonderful (and crazy) people I met during those six months in Dundee: thank you guys. I hope I'll see you soon.

Samu

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 About this chapter

The main goal of the ophthalmology research community is to improve the understanding of the underlying causes of eye-related medical conditions [1] and provide a solution for them. Nowadays the estimated number of visually disabled people world-wide is about 285 millions [2], this indicates visual impairment as a major health issue. The newest advances in both computational power and image processing techniques give the opportunity to the clinical world to meet the eye-research community. The main causes of global blindness are glaucoma, cataract, macular degeneration and diabetic retinopathy (DR); most of the study on automatic lesion detection is oriented on the last one. Studies revealed how early detection of those pathologies could significantly influence their treatment and consequently improve the health of the patients [1]. Retinal morphology can also provide helpful information for the detection of cardiovascular and cerebrovascular diseases; in those cases too an early diagnosis allows preventive measures. Novel image analysis and machine learning techniques for, vessel segmentation [3, 4], artery/vein classification [5] or vessel width estimation [6] help clinicians to provide a better diagnose and follow-up of the majority of retina-related deceases. Getting data to help researchers to develop and improve retinal image analysis techniques is a mammoth task and it is the main goal of the presented work.

In this chapter we will introduce terminology, main ideas and purposes of the presented work, but, first of all, we are going to present a brief recap of eye-anatomy main features.

1.2 Simple anatomy of the eye

When looking into someone's eyes several structures are easy to be recognised[1] (Figure 1.1). The *pupil* (1) is a black-looking aperture that allows light to enter the eye. Pupil's size is controlled by a surrounding coloured circular muscle called *iris* (2). The iris controls pupils dimensions according to external illumination conditions. Iris and pupil are covered by a transparent external lens called cornea (3), that together with the *crystalline lens* (4) let us see sharp images. The white supporting structure is called *sclera* (5). An adult human eye is a slightly asymmetrical sphere with diameter

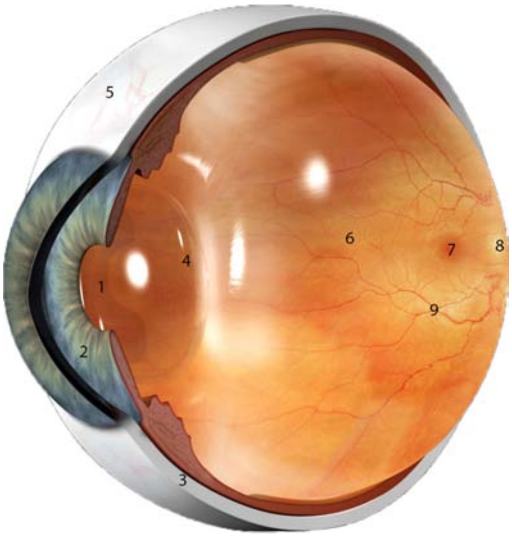


Figure 1.1: Section of the adult human eye[2]

around $24 - 25mm$ and volume of about $6.5cm^3$. As far as we are concerned, it is possible to distinguish between three layers: the external one, formed by cornea and sclera, the intermediate, where the iris is placed, and the internal membrane, the *retina* (6). In this work we will, obviously, mainly focus on the last one. Keeping the adequate oxygen supply to the retina is critical for its health and function [7]. Oxygen can reach the retina in two ways: mainly through a vascular layer lying under the sclera, called *choroid*, and secondarily ($\sim 35\%$) through the *retinal vasculature* (9). In the centre there is the head of the optic nerve (8), a yellowish circular or oval structure from which the four major vascular arcades come out. Its area is usually about $3mm^2$ and, because of its shape, this structure is usually called *Optic Disc* (OD). As we said, the retina has a complex and articulated blood vessel network; from the centre of the OD four arcades start. Two of those are veins and two are arteries. For both arteries and veins, each arcade has a parabola-like shape and a principal direction: one goes to the right and one to the left of the OD. Roughly 2.5 OD diameters left of the disc there is a reddish oval-shaped vessel-free spot: the *fovea* (7). In retina fundus photography it is common to see a small white point placed in the centre of the fovea, the so-called *macula*. A flat projection of the total retina could be a disc with a diameter of about $30 - 40cm$ and thickness around $0.5mm$. Simplifying, the main function of the retina is the same as that of the film in an analog camera; when the light, travelling through the thickness of the eye, reaches the membrane the lying photoreceptor cells (*rods* and *cones*) start a biochemical message that later on becomes an electrical signal and reaches our brain through the optic nerve ganglions.

[1] Credit for this description goes to Webvision [8], the first online textbook by Helga Kolb [http://webvision.med.utah.edu].

[2] The eye section in Figure 1.1 has been rendered by Jojin Kang, a CGSociety artist [http://www.cgsociety.org].

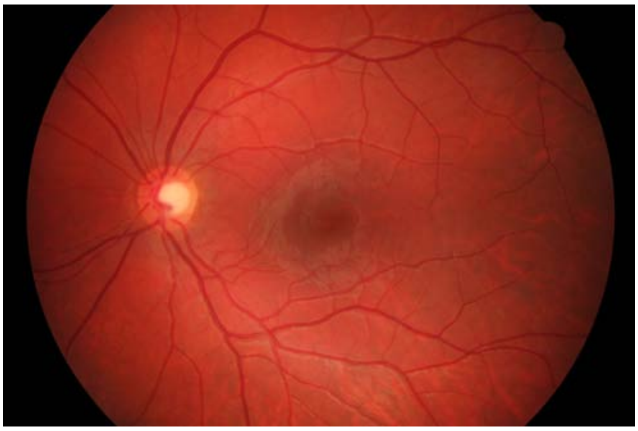


Figure 1.2: A human healthy retinal fundus image[3].

1.3 What is Retinal Image Analysis?

We define *Retinal Image Analysis* as the interdisciplinary field that aims to develop computational and mathematical methods for solving problems and increasing the knowledge of the human retina.

As we can understand from the previous section, to allow rods and cones to capture the light from the outside, the ocular structure lying between cornea and retina must be optically transparent [9]. This gives the opportunity to view the retinal tissue from the outside noninvasively (Figure 1.2). The first ophthalmoscope was invented around mid-19th century and retinal inspection quickly became routine for ophthalmologists. In 1910 Gullstrand[4] developed the concept of fundus camera and for his invention he was later awarded with a Nobel prize. In the last century retinal imaging has rapidly developed and it is now widely used for the diagnosis of many diseases. The new-generation high resolution fundus cameras let ophthalmologists take extremely detailed retinal images, in the very last year even some smartphone-based solutions have been invented. Fundus photography is not the only retinal imaging technique, other modalities exist, like *scanning laser ophthalmoscopy (SLO)* and *fluorescein angiography*. In this study we only work with digital colour fundus images.

1.4 Validation in Retinal Image Analysis

In a medical image processing environment, validation can be defined as *the process of showing that an algorithm performs correctly by comparing its output with a reference standard* [11]. Validation of a medical image processing method is the best way the research community has to evaluate its performance or limitation and to understand its possible application in a clinical practice. As outlined in [12], a consequence of the rise of medical image analysis techniques demands the definition

[3] Image 14, HRF database, healthy subsection [10].

[4] Allvar Gullstrand (5 June 1862 - 28 July 1930), professor of eye therapy and optics at the University of Uppsala, was a Swedish ophthalmologist and optician.

of standard validation methodology and the design of public available datasets. The presented work aims to meet the second need. Four main types of data sets can be distinguished depending on the availability of the ground truth (GT). The GT could be perfectly known (the so-called *absolute GT*), can be computed from the data, can be given by observers or can come from some a priori clinical knowledge. Inter-observer variability and wrong assumption could heavily affect the reliability of the last two cases. For instance, it is very unlikely that different experts could present 100% agreement estimating the same vessel widths or graduating the same corneal nerves tortuosity, particularly in case of large datasets. The use of physical phantoms could allow to obtain a reliable GT. While this is a common practice in radiologic or MRI environment, to our viewpoint it is not easily applicable in retinal fundus imaging. To our best knowledge, the studies presented in literature which have implied physical eyeball phantoms so far were mainly addressed to surgery applications. Our main goal is to simulate an *absolute GT* for retinal fundus image (RFI) analysis applications. In order to clarify how this could be helpful for the validation of new algorithm let us refer to a comprehensive analysis of the state-of-the-art challenges about validation in retinal image processing by Trucco and Ruggeri [11]. Validating an automatic retinal image analysis technique implies the following four points:

1. the selected **database** must be a representative sample of the target population;

2. the **annotation collection**, if needed, must be reliable and with low inter-observer variability;

3. it is necessary to **run algorithms** on the sample images;

4. the **comparison with the reference standard** should assess the agreement between ground truth and the output of the algorithm.

In relation to point 1, the presented work describes a method that allows a user to generate novel realistic colour retinal fundus images. Those images are similar in characteristics to a given dataset but their textural anatomical features can be modified in order to simulate a wide range of parameters as well as different populations. About points 2 and 4, this method can potentially provide a GT with 100% reliability and free from inter-operator variability; in fact the synthetic RFI is actually generated at a later stage according to the GT. If the tested algorithm is, for instance, an OD detector, the user will be provided with OD boundary map, coordinate of its centre and a complete description of its morphology.

The development of this work has required a large use of patch-based approaches and a novel parametrical model for the synthesis of OD has also been proposed. With the presented software a user can now reproduce RFI with resolution and Field-of-View (FOV) consistent with the *High-Resolution Fundus Image Database* (HRF) [10]. This work mainly focuses on the synthesis of non-vascular regions, this is complemented by a work-in-progress parallel study on the generation of vessel tree structure and texture. The method is implemented in MATLAB$^{®}$ and a graphical user interface (GUI) guides the user into the workflow, for further information see the Appendix on page 65.

1.5 Aim and structure of the Thesis

The presented work aims at generating realistic retinal fundus image phantoms similar in characteristics to a given dataset. Along with the synthetic images a reliable absolute ground truth is provided as well. The final goal will be the public release of a new database containing synthetic retinal images supplied with OD and macula centre coordinates, vessel binary maps and width at any position and artery/vein GT maps. Our synthetic data can find multiple applications, first of all they have an immediate use in algorithm test and validation where the necessity of a reliable GT is a well documented evidence. On the other hand, allowing to manually modify retinal textural and anatomical characteristics, this work fits very well with the requirements to become a major tool for the training of both the current and the next generation of ophthalmologists.

This thesis is organised as follows. **Chapter 2** provides a general overview of previous works on medical imaging phantoms, gives some examples in ocular surgery field and presents some jointed computer vision-graphics techniques. **Chapter 3** describes in details all the proposed algorithms analysing their implementations and the occurred problems. **Chapter 4** provides a wide range of synthesised images with variable textural characteristics and shows the results obtained from the conducted quantitative tests. Finally in **Section 5** we present our conclusions and some hints for future works.

1.6 Chapter Summary

In this chapter we introduced the main concept behind the presented method and we also motivated it. Before getting into details, the next chapter will provide a brief overview of some eye-related image analysis (and synthesis) technique.

Chapter 2

Related Work

2.1 About this chapter

In this chapter we clarify the necessity of imaging phantoms and we point out pros and cons of both physical and digital approaches giving an overview of the related most-known works proposed in literature.

2.2 Phantoms in medical imaging

As we clarified in Section 1.4, the GT reliability is a well-known dilemma. Validation using *in vivo* collected data is difficult because there is no direct information available to corroborate the algorithms. Manual labelling is a common solution, but it is hazily affected by inter-observer variability [13]. In radiology, magnetic resonance (MR) and computed tomography (CT) physical phantoms are extensively used. They allow to control the object shape or dimension and material properties. The first example is the well-known Shepp-Logan phantom [14]; this model was created in 1974 as a standard for CT but it is also often used for validating MR reconstruction algorithms [15]. It contains ten ellipses of different size and a material that tries to simulate the x-ray attenuation properties of a human head. It becomes a standard test image for testing numerical accuracy of two-dimensional reconstruction algorithms. It is so popular that MATLAB® provides a built-in function for its generation. In Figure 2.1 is depicted an example generated with the MATLAB® command `phantom`. Later more complex and anatomically realistic phantoms have been created, an example is the Hoffman brain phantom [16], developed to simulate the cerebral blood flow and designed for positron emission tomography (PET) studies.

Physical phantoms allow reliable testing in real-world data acquisition, but their use presents some relevant issues. Building a physical phantom could be a complex and expensive task and their stability makes it also difficult to change their shape and properties materials. In the last few years digital phantoms have been proposed to address those issues [17, 13] and this is exactly the presented work domain. The advantages of digital phantoms are many, first of all every parameter can be tuned in order to simulate different pathology or to change anatomic shapes. Another relevant pro is that with simulated data possible sources of error are easy to control. Artefacts or signal-to-noise ratio (SNR) can be modulated so that it is possible

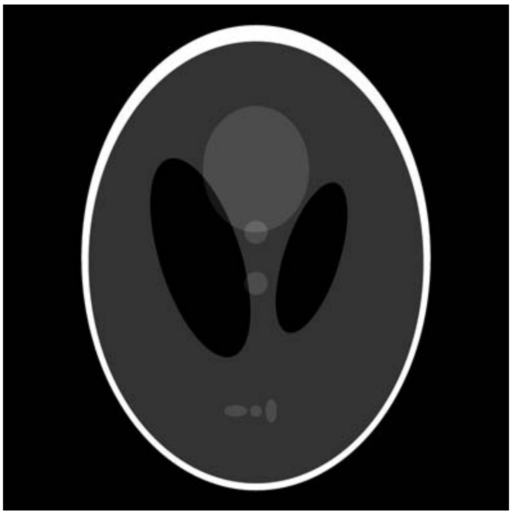


Figure 2.1: The Shepp-Logan phantom [14].

to better evaluate the noise robustness of the tested method. Last, but not least, development and use of digital phantoms is very cheap because it does not involve raw materials nor expensive equipment, like MR scanners or, in our case, fundus cameras. In the next section we are going to analyse the main works on synthetic eyes proposed in literature.

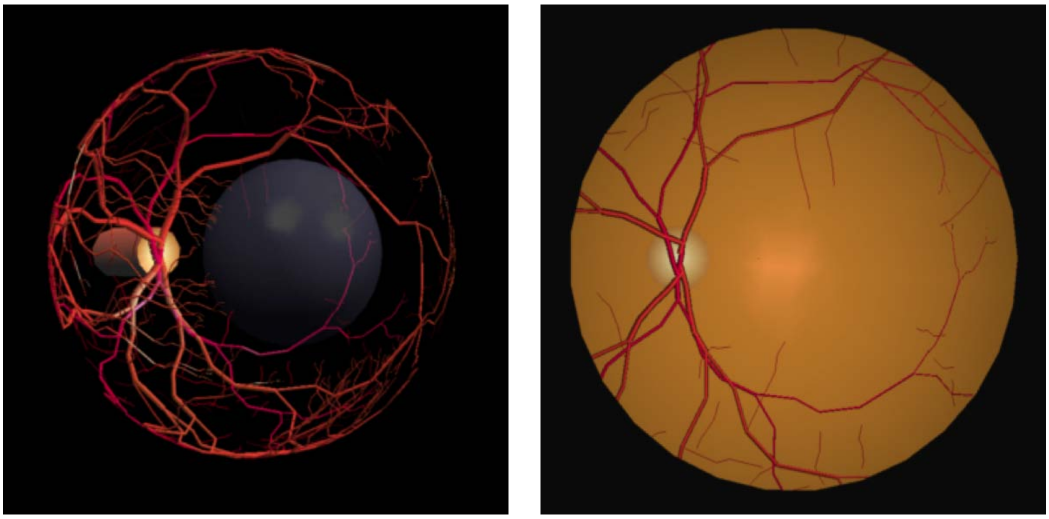
2.3 Previous works on eye modelling

2.3.1 Retinal vascularisation

Most of the previous works on human eye modelling are addressed to surgery simulation [18] or to global eyeballs image rendering [19]. To our best knowledge there are no previous works on the synthesis of RFI phantom.

The first detailed eye rendering was published in 1994 by a research team of The University of Auckland, New Zealand, in collaboration with the McGill University of Montreal, Canada [18]. In this work an anatomically meticulous 3D model of human eye with surrounding face has been developed for a use in surgical simulation. A surgical virtual environment has been developed both to aid surgeons during actual operation and provide realistic training. This work has also a micro-surgical robotic system teleoperated by the surgeon during the simulation. In order to provide the surgeon with a feedback that is as realistic as possible, finite element models and the large deformation theory with orthotopic nonlinear material properties have been used. The authors were really keen on rendering every details; they provide a convincing sclera, cornea, iris and even eyelids, but the most relevant part for our purpose is their representation of retina and blood vessels (Figure 2.2).

In Figure 2.2a is depicted a retinal blood vessel network projected onto a sphere containing a central black globe, modelling the fovea, and on which a yellowish bordering cylinder represents the optic nerve head. A recursive algorithm generates the vessel tree which, as the authors declared, is subject to a number of parameters



(a) 3D retinal surface rendering with fovea, OD and vessel network.

(b) 2D RFI render.

Figure 2.2: Results presented by Sagar, Bullivant et al. [18].

that allow to control features like mean branching angles, vessel radii and undulations, junctions and bifurcations. A repulsion factor prevents the fovea to be populated by vessels. In Figure 2.2b a first attempt of synthetic RFI is depicted; the image looks reasonable at a first glance, but after a scrupulous analysis noticeable shortcomings are easy to recognise. Comparing Figure 2.2b and the real image depicted in Figure 1.2 it emerges how the colour in non-vascular regions is unlikely smooth while it should have a texture and a non-homogeneous illumination. The fovea is rightly vessel-free, but it appears brighter than the background while we know that it is usually darker. We can also see that the OD does not present the almost white internal cup-like area. The level of branching reached by a real vasculature network should be, in our opinion, higher than the one represented there and there is not any texture difference between arteries and veins. By the way, taking into account that first of all rendering a convincing RFI was not on the author's goals list and second that that paper was published in 1994, those results are a really good pioneering starting point.

2.3.2 Optic Disc Texture and Appearance

In order to generate convincing RFI phantoms, a preliminary study on retinal structures appearance has been conducted. In this regard, to our best knowledge, the literature lacks in detailed description of the background, while a detailed OD description is easy to find in many works. As previously mentioned (Section 1.2) the OD is the extremity of the optic nerve and also the entrance and exit site of retinal vessels [20]. Its shape is approximately elliptical, with a vertical principal axis. There is often a brighter region, called *pallor*, which usually includes a centred small depression called *cup*. In RFI the appearance could vary a lot, the OD size is strictly correlated with the technical specifications of the fundus camera (e.g. resolution and FOV). The OD is usually brighter in its side close to the fovea, looking at different coloured RFIs we realise that it could appear in general as a yellowish, pink or even almost red circle sometimes surrounded by a bright area caused by *peripapillary atrophy*. The OD cup in glaucomatous patients has a different appearance, glaucoma increases the intra-ocular pressure generating larger cups. A common used measure for diagnosing and monitoring glaucoma is the *cup-to-disc ratio* (CDR), that is the ratio between disc and cup vertical diameters. This variability in appearance makes the development of a realistic OD a difficult task, we will see in Section 3.3.2 the proposed model-based method. Lowell et al. in [20] describe an OD localisation technique which using the *full Pearson-R correlation* index tries to estimate its centre by matching a template function with a grey-level intensity retinal image. The template surface is a *Laplacian of Gaussian* (LoG) (Figure 2.3) with a vertical channel in the middle corresponding to the major vessel area. The 2-D LoG centred in zero with standard deviation σ has the form described in Equation 2.1.

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

The proposed texture synthesis model is different (and it will be discussed in Section 3.3.2), but this work gave us the idea of using a parametric model for the synthesis of the OD.

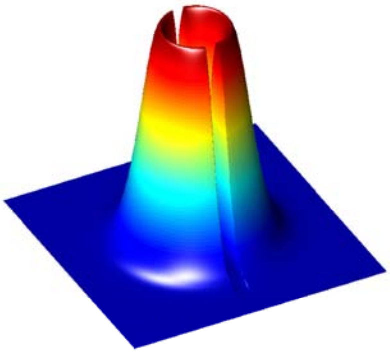


Figure 2.3: Laplacian of Gaussian surface with vertical channel.

2.3.3 Eyeball rendering

The last briefly mentioned work was presented in 2009 by François et al. [19] and proposes an image-based method for estimating iris morphology and rendering of convincing virtual eyes. Apart from the astonishing results they achieve (Figure 2.4), this method presents the idea of learning parameters from real images and use them for their rendering. On the same line, in Section 3.3.2 we present a model-based approach for the synthesis of realistic OD with parameters learned from real data.

Figure 2.4: François et al. [19] 3D eye rendering.

2.4 Computer Vision and Graphics techniques

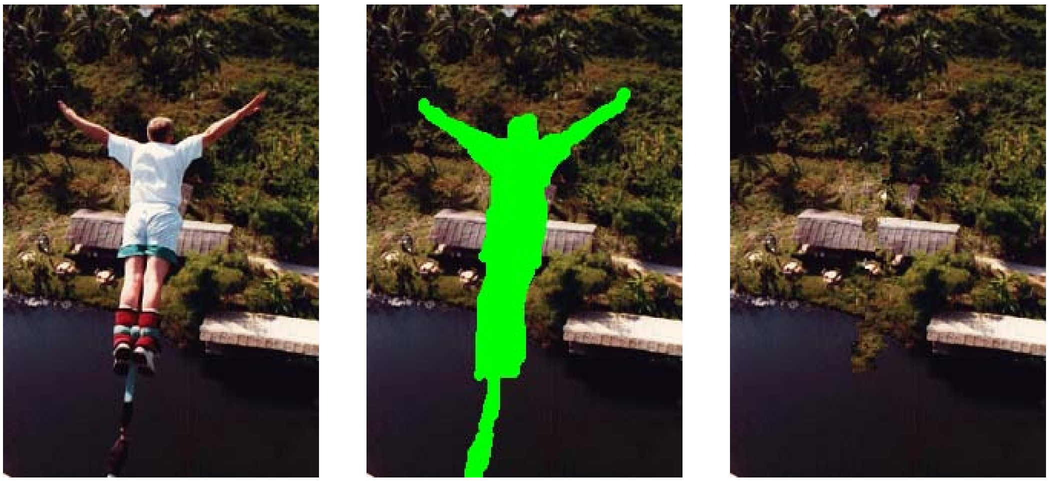
2.4.1 Image Inpainting

During the development of this work we came across more than one obstacle and in order to overcome them a wide use of Computer Vision and Graphics techniques has been done. Among them, image *Quilting* and image *Inpainting* have been particularly helpful and inspiring. A description of those methods is provided below.

The presented study mainly concerns the texture synthesis of non-vascular regions in RFI and real images have been our starting point. Looking at Figure 1.2 we can see how in an RFI is easy to winnow between the background and the elements of the foreground (OD, vessel tree). A question immediately arises: *how can we perform a background estimation?* We may notice that, at this level, the fovea is simply considered as a shade of the background. The basic idea is to think at OD and vessels as some scratch we want to remove from an old photography or, more in general, as large unwanted objects we want to delete. This is a well-known problem called *image restoration*. Many techniques have been proposed to address this challenge [21, 22, 23], generally they can be categorised into two classes: *Diffusion-Based* and *Exemplar-Based*. Methods in the first class try to fill a target regions propagating linear structure with equal intensity value called *isophotes*. Those methods are inspired by the differential equations of physical heat flow. A well-known example is the popular *Image Inpainting* by Bertalmio et al. [23], in which the isophotes arriving at the target regions boundaries are propagated inside keeping the same angle and curving to prevent them from crossing each other. Diffusion-based techniques work convincingly as image restoration, but their drawback is the introduction of some blur in the inpainted region. On the other hand, Exemplar-Based approaches relies on the idea of replacing a target region by sampling an copying small parts of texture from a source, this concept originates from the already mentioned image Quilting technique by Efros and Freeman [24] which will be discussed later (see Section 2.4.2). The algorithm we selected for our purpose is the very famous Exemplar-Based technique proposed in 2004 by Criminisi et al. [21]. This technique combines the strength of isophotes propagation, with the texture coherence of a classic patch-based approach. We can sketch the algorithm as follows[5].

Given a input image I (Figure 2.5a) a target region Ω to be filled is (manually)

[5] A MATLAB® function implementing Criminisi's technique by Sooraj Bhat is available at http://www.cc.gatech.edu/∼sooraj/inpainting/



(a) Original image. **(b)** Target region manual segmentation. **(c)** Inpainting resut.

Figure 2.5: Criminisi's Inpainting Technique results[6] [21].

selected (Figure 2.5b) and its boundaries $\delta\Omega$ are evaluated. Let us define Φ as the source region where $\Phi = I - \Omega$. The template window Ψ size has to be specified, the authors suggest to choose a dimension slightly larger than the largest distinguishable texture element in the source region. Once those parameters are determined the algorithm iterates the following three steps until all the target region has been completely filled.

1. *Computing Patch Priorities*, the algorithm follows a best-first filling strategy and a priority is assigned to every patch Ψ_p centred at the point p for some $p \in \delta\Omega$. The priority term $P(p)$ is defined as $P(p) = C(p)D(p)$, where $C(p)$ is the confidence term and $D(p)$ the data term; $C(p)$ intuitively is the amount of reliable information surrounding the pixel p while $D(p)$ is a function of the strength of isophotes hitting the front $\delta\Omega$.

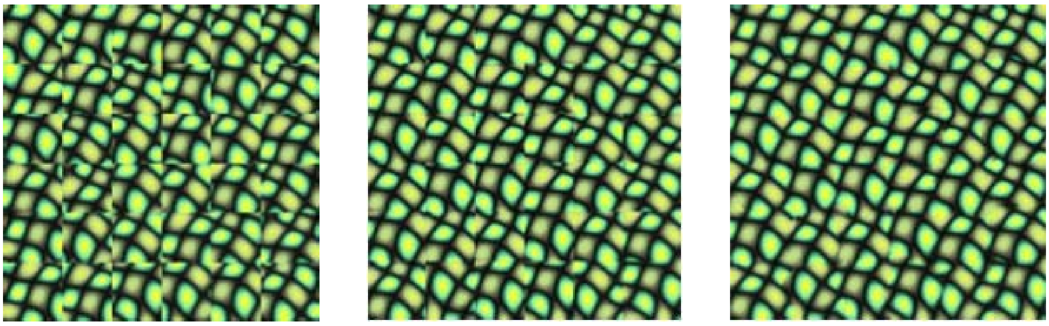
2. *Propagating Texture and Structure Information*, assuming the patch $\Psi_{\hat{p}}$ as the one with maximum priority, it is now filled with a patch $\Psi_{\hat{q}}$ sampled from Φ defined as

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}} - \Psi_q) \tag{2.2}$$

   where $d(A, B)$ denotes the sum of squared differences (SSD) of the already filled pixel in two patches A, B.

3. *Updating Confidence Values*, after $\Psi_{\hat{p}}$ has been filled, $C(p)$ is consequently updated following the new configuration.

In Figure 2.5 the performance of the algorithm can be visually estimated. Figure 2.5a depicts the original image, Figure 2.5b shows the selected area to be inpainted and finally in Figure 2.5c we can see the result of Crimini's technique. The bungee



(a) Random Block Placement. **(b)** Neighbouring blocks constrained by overlap. **(c)** After minimum error boundary cut.

Figure 2.6: The Quilting Algorithm[7].

jumper has been completely removed but a little unwanted "*peninsula*" in the body of water has been created. Nevertheless we will see in Section 3.3.1 that this algorithm has good performances in our application as well.

2.4.2 Image Quilting

As we mentioned, Criminisi's inpainting technique relies on the idea of using patches to fill a target region. This concept is inspired from the well-known image Quilting technique proposed by Efros and Freeman in 2001 [24]. In our work large use of a modified version of that patch-based texture synthesis technique has been done (see Section 3.3.1). The final aim of the original method is the generation of a new image synthesised by stitching together small patches belonging to another image. Let us first define the *unit of synthesis* B_i, a square block of user-defined size belonging to S_B, the set of all the overlapping blocks in the input image. The final result will be obtained following the idea sketched out in the next three basic steps.

1. If we try to fill an area simply with blocks randomly taken from S_B (see Figure 2.6a); the result could be reasonable, but with too strong edges between each block. A good match of randomly taken blocks is very unlikely, so let us try a simple trick.

2. We now introduce some overlap in the *North* and *West* side of every patch and we add the following constraint at the process of patch selection: for every new block B_{i+1} we will look in S_B for a tile that agrees with its neighbour B_i along the overlap. The result is showed in Figure 2.6b, it has a relevant improvement but the edges are still visible and the black lines do not continue smoothly from one block to another. So we introduce the last trick.

3. We may notice that at this point we still have an unresolved question: *which is the best behaviour the algorithm could have in the overlap area?* A possible

[6] Figure 2.5 belongs to the original *Region Filling and Object Removal by Exemplar-Based Image Inpainting* by Criminisi et al. [21].

[7] Figure 2.6 belongs to the original *Image Quilting for Texture Synthesis and Transfer* by Efros and Freeman [24].

solution to this doubt is the so-called *Best Boundary Cut* (BBC) that makes the transition from B_i to B_{i+1} visually pleasant. This cut could be evaluated using a dynamic programming technique or also with a shortest-path method like the Dijkstra's algorithm (mainly used in graph theory). Let us choose the first solution introducing the *Minimal Cost Path*. If B_1 and B_2 are two blocks overlapping along the W edge in the regions B_1^{ov} and B_2^{ov}, we define an *error surface* $e = (B_1^{ov} - B_2^{ov})^2$. In order to find the BBC we traverse e for $i = 2, \ldots, N$, where N is the number of rows, and compute the *Cumulative Minimum Error* $E_{i,j} = e_{i,j} + min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$. At last the BBC is identified scanning row-wise the obtained result and selecting for each row its minimum value. The original paper does not specify the behaviour of the cumulative error formula in the first and last column, where $E_{i-1,j-1}$ or $E_{i-1,j+1}$ are out of bound. To bypass that problem we assume that $j = 2, \ldots, M-1$ where M is the number of columns. Once the BBC is evaluated the patch is stitched to its neighbour according to that.

In the original paper a wide number of results is available (see [24]), from their analysis we can conclude that the algorithm is particularly effective for semi-structured textures and performs good on stochastic textures as well. As we will widely discuss later (see Section 3.3.1), this makes the Quilting technique particularly suitable for the synthesis of RFI backgrounds.

2.4.3 Colour Similarity Measurement

In this project the concept of *Colour Similarity* has widely been explored. The idea is to make a quantitative measurement of distance between colours in a camera-independent colour-space. Otherwise that notion could be just explained with a sensation. Achieving this goal is not easy because of the different spectral sensibility the human eye has to different colours [25]. The International Commission on Illumination (CIE) stated a standard distance measure called ΔE of which many definitions have been proposed over the years. During the development of the current project the definition proposed in 1994 [26] has been taken into account. To measure ΔE^*_{94}, a system of equations that predicts colour differences working in the non-uniform (L^*, a^*, b^*) colour space is needed. Using two colours (L^*_1, a^*_1, b^*_1) and (L^*_2, a^*_2, b^*_2) the CIE94 colour difference can be stated with the following equations.

$$C^*_1 = \sqrt{a^{*2}_1 + b^{*2}_1} \tag{2.3}$$

$$C^*_2 = \sqrt{a^{*2}_2 + b^{*2}_2} \tag{2.4}$$

$$\bar{C}^*_{ab} = \sqrt{C^*_1 C^*_2} \tag{2.5}$$

$$S_C = 1 + K_1 \bar{C}^*_{ab} \tag{2.6}$$

$$S_H = 1 + K_2 \bar{C}^*_{ab} \tag{2.7}$$

$$\begin{Bmatrix} \Delta L^* \\ \Delta a^* \\ \Delta b^* \end{Bmatrix} = \begin{Bmatrix} L_1^* - L_2^* \\ a_1^* - a_2^* \\ b_1^* - b_2^* \end{Bmatrix} \tag{2.8}$$

$$\Delta C_{ab}^* = C_1^* - C_2^* \tag{2.9}$$

$$\Delta H_{ab}^* = \sqrt{\Delta a^{*2} + \Delta b^{*2} - \Delta C_{ab}^{*2}} \tag{2.10}$$

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2} \tag{2.11}$$

Where the weights are set according to the table below.

S_L	1
K_1	0.045
K_2	0.015
k_L	1
k_C	1
k_H	1

The non-linear relations of this colour-space tries to simulate the non-linear behaviour of the human eye. This measure seemed to work very well in detecting similar colours, but its major drawback is the necessary long elapsed cpu-time. As we can see from the equations above, in fact, ΔE_{94}^* evaluation needs many preliminary steps, and a conversion of the images from the standard (r, g, b) in the (L^*, a^*, b^*) colour space is needed. This transformation is a cpu-intensive task as well. Because of this reason ΔE_{94}^* has not been the selected colour distance measure.

After a scrupulous research we chose the colour metric presented by Guobing et. al in 2011 [27]. This colour distance method, originally applied for real-time bleeding detection in wireless capsule endoscopy, works in (r, g, b) colour space and, as the authors claimed, leads their bleeding detection algorithm to a sensitivity and a specificity of 97% and 90% respectively. This measure index relies on the idea that any pixel could be represented as a 3-D vector in a Cartesian coordinate system (see Figure 2.7) in which it is possible to distinguish between an *amplitude* similarity coefficient ($n_{<x,y>}$) and a *chroma* similarity coefficient ($\gamma_{<x,y>}$). Taking into account the two vectors $C_1(r_1, g_1, b_1)$, $C_2(r_2, g_2, b_2)$ represented in Figure 2.7 and defining $|C_i| = \sqrt{(r_i^2 + g_i^2 + b_i^2)}$ the amplitude similarity index can be stated as follows (Equation 2.12).

$$n_{<C_1,C_2>} = 1 - \frac{||C_1| - |C_2||}{|C_2|} \tag{2.12}$$

Intuitively this measure is a grey level colour similarity index, the smaller $n_{<C_1,C_2>}$ is, the less the two colours intensity is similar. On the other hand we observe that in case of a perfect match $C_1 = C_2$ we have $n_{<C_1,C_2>} = 1$ (i.e. the maximum value). The chroma similarity index, instead, can be intuitively defined as a function of the angle (α) between two 3-D vectors in the (r, g, b) system (see Figure 2.7). A formal statement of this measure is expressed below (Equation 2.13) where the notation $x(i,j,k) \cdot y(i,j,k) = x_i y_i + x_j y_j + x_k y_k$ is intended as the vectors inner-product.

$$\gamma_{<C_1,C_2>} = cos(\alpha) = \frac{C_1 \cdot C_2}{|C_1||C_2|} = \frac{r_1 r_2 + g_1 g_2 + b_1 b_2}{\sqrt{(r_1^2 + g_1^2 + b_1^2)}\sqrt{(r_2^2 + g_2^2 + b_2^2)}} \tag{2.13}$$

We may notice that the bigger the included angle α (with $\alpha \in [0, \pi]$) is, the smaller the value of $\gamma_{<C_1,C_2>}$ becomes. This means that a high difference in directions of the 3-D colour vectors corresponds to a small value of the chroma similarity index. When $C_1 = kC_2$, where k is a proportional factor, the two directions are exactly the same and $\gamma_{<C_1,C_2>}$ reaches its maximum in 1. The paper describes how to use those colour similarity indexes in a classification algorithm, but this is not necessary for our purposes. We will describe in Section 3.3.1 how those two measures have been combined together to fulfill our requirements.

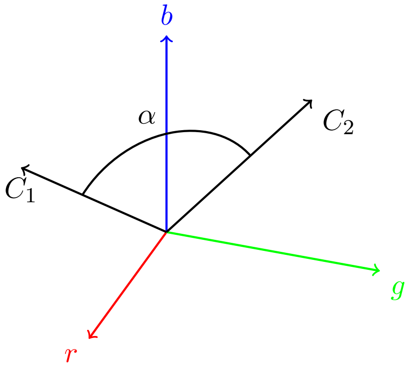


Figure 2.7: Three pixels in a (r, g, b) Cartesian colour space.

2.5 Chapter Summary

In this chapter we had an overview of which works inspired us during the development of our project, we introduced the main concept of image *Phantoms* and *Eye-Modelling*. We also sensed the possible *texture synthesis* related issues. Both *patch-based* and *model-based* approaches have been introduced and a good description of the most relevant Computer Vision and Graphics techniques has been done. In the next chapter a comprehensive and detailed explanation of the presented method is proposed.

Chapter 3

Proposed method

3.1 About this chapter

In this chapter we will start with a description of both materials and software used, then we will accurately describe the proposed algorithm and the main implementation issues. We will eventually outline the workflow to follow for the synthesis of a novel RFI phantom. To achieve our final goal we divided the global work into two separate stages: the generation of vascular and non-vascular regions. As already mentioned, the presented work focuses on the generation of non-vascular regions, so henceforward we will not take into account any vessel tree contribute. The method will be described in the same order as it has been developed, we will first see background and fovea generation algorithm and then the implemented OD texture synthesis technique.

3.2 Materials and Softwares

3.2.1 High-Resolution Fundus Image Database

As we mentioned, with the presented work we are able to synthesise convincing RFI phantoms similar in characteristics to a certain dataset. Because of its high resolution and wide FOV, we selected the recently released *High-Resolution Fundus Image Database* [10]. HRF[8] was mainly created to support comparative studies on automatic segmentation algorithms on retinal images. HRF is made of 45 images taken at the Eye Ophthalmology Clinic Zlin, Czech Republic. It contains three subsets of fundus images, with 15 images each. The first set contains RFI without any retinal condition, the second has images taken from patients with glaucoma in advanced stage and the third subset presents RFI acquired from patients with DR; they have recognisable haemorrhages, exudates, bright lesions and spots after laser treatments. All the images were acquired with the mydriatic fundus camera CANON CF-60 UVi equipped with CANON EOS-20D digital camera with a 60-degree FOV, standard mydriatic drops were used to dilate subject's pupils. The images are 3504 × 2336 pixels, they are encoded with 24-bit per pixel and stored in standard JPEG format with low compression rates. For each image a binary mask of the FOV is also provided. In every image the vessel tree has been manually segmented by three experienced ophthalmologists of the clinic using ADOBE Photoshop CS4. An example of HRF image is depicted in Figure 1.2. As a matter of fact some small

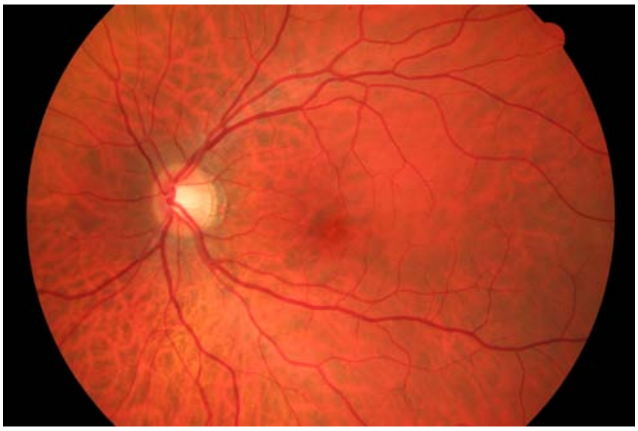


Figure 3.1: Healthy HRF retinal image with choroidal vessels in the background.

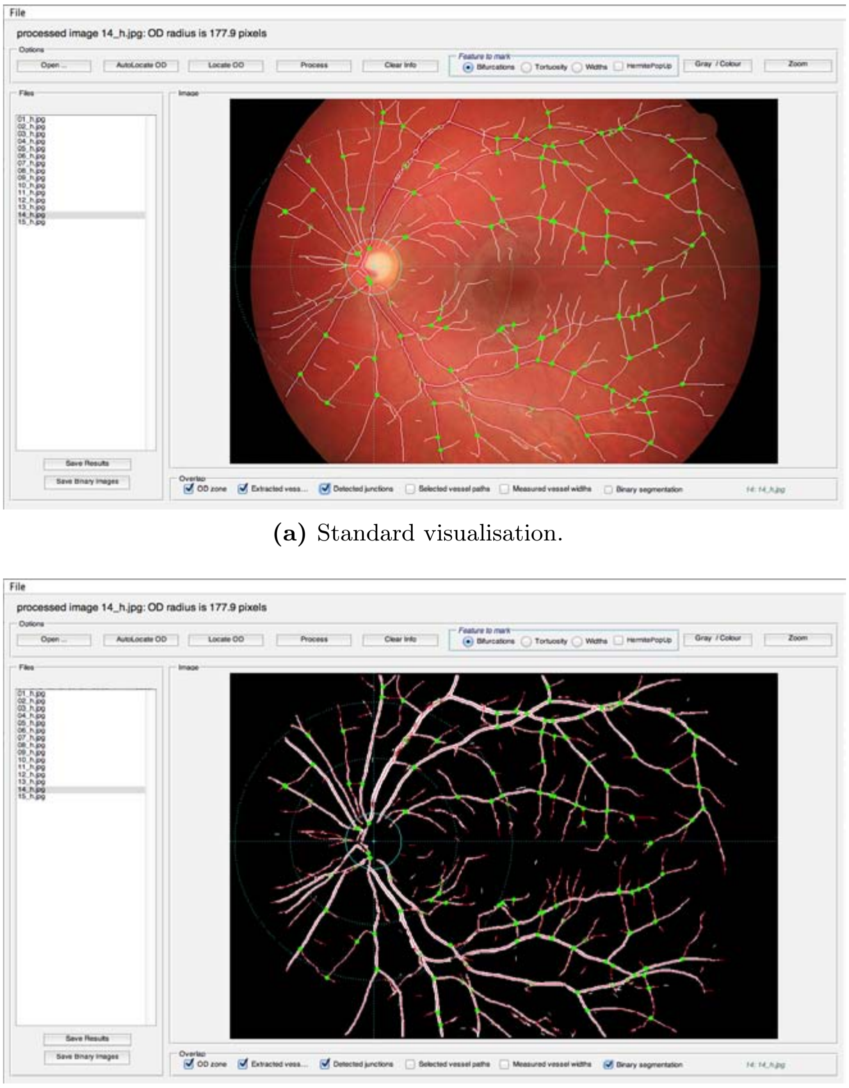
inaccuracies in the provided manual vessel GT are recognisable. Some vessel edges have a non-physiologic shape and background pixels are sometimes included between two vessels in the same junction. On the HRF database website a spreadsheet which presents all the manual OD annotations is also provided.

3.2.2 VAMPIRE software suite

VAMPIRE (Vessel Assessment and Measurement Platform for Images of the REtina [28, 29]) is a software suite born in 2011 and in continuous development. It is designed for semi-automatic retinal image analysis and annotation. It mainly consists in two tools (both available with a user-friendly MATLAB® GUI): VAMPIRE himself and the parallel VAMPIRE-Annotation Tool. Both those softwares have been extremely helpful during the development of the presented work.

VAMPIRE 2.0 allows the user to load a set of RFI of arbitrary size and provides algorithms for robust automatic detection of retinal landmarks, as OD centre, vessel network, bifurcation and junction, and retinal parameters quantification, such as vessel width and tortuosity. A screen capture is presented in Figure 3.2. Currently VAMPIRE implements the OD localisation algorithm proposed by Giachetti et al. in [30]. It is based on a very simple concept: the OD is an ellipse shaped structure brighter than the local background with blood vessels in its central part. The algorithm mainly consists in four steps: *a*) a grayscale conversion is applied to the image, *b*) a vasculature map is generated with a quick segmentation technique, *c*) all the vessels are inpainted using and iterative median-based method and *d*) a multiresolution ellipse detection scheme is then applied. The OD contours are so eventually identified. The last VAMPIRE version implements a vessel detection algorithm which is a version of the well-known Soares et al. technique proposed in [4]. VAMPIRE uses this algorithm to generate tree-like representations of the vasculature network as a pre-processing for measurements, but the software also lets the user save the vessel tree as binary map (see Figure 3.2b). The VAMPIRE-Annotation Tool is a software for manual annotation of retinal features. It basically allows the

[8] The HRF database can be downloaded at http://www5.cs.fau.de/research/data/fundus-images/



processed image 14_h.jpg: OD radius is 177.9 pixels

Feature to mark: ● Bifurcations ○ Tortuosity ○ Widths □ HermitePopUp

Overlap: ☑ OD zone ☑ Extracted vess... ☑ Detected junctions □ Selected vessel paths □ Measured vessel widths □ Binary segmentation

(a) Standard visualisation.

processed image 14_h.jpg: OD radius is 177.9 pixels

Feature to mark: ● Bifurcations ○ Tortuosity ○ Widths □ HermitePopUp

Overlap: ☑ OD zone ☑ Extracted vess... ☑ Detected junctions □ Selected vessel paths □ Measured vessel widths ☑ Binary segmentation

(b) Vessel map visualisation.

Figure 3.2: VAMPIRE working on HRF database.

user to manually localise retinal landmarks, as OD, fovea and vessel junctions, and to measure the vessel width at specified locations. The vessel tortuosity annotation feature is currently a work-in-progress. The VAMPIRE-Annotation Tool provides also the same automatic OD detection already described for the main VAMPIRE project. Both VAMPIRE and VAMPIRE-Annotation Tool provide many other interesting features (like fractal analysis or branching coefficients quantification), but in the development of our work we were just interested in a reliable vessel segmentation technique (see the quantitative tests described in Section 4.4 on page 57) and in an easy retinal landmark manual annotation software (see the co-registration step in Section 3.3.1).

3.3 Description of the Method

3.3.1 Background generation

In order to generate convincing RFI phantoms, we started from the observation of real images. At a first glance, retinal images (as the one in Figure 1.2) are showing three obvious elements: Optic Disc, Vessel Network and Fovea. All of them are surrounded by an orange-red background. Narrowly watching a fair number of images the observer realises that the background could have different textures and its average illumination is non-homogeneous; it usually follows the vessel tree, being brighter in the vascularised areas and darker in the fovea and towards the extremities. Intuitively the average colour distribution in a RFI could remind us of a *horseshoe*. Choroidal vessels are also sometimes visible in transparency (see Figure 3.1), this effect creates a real texture, very different than a mere smooth colour fill. In order to synthesise a new background preserving both non-uniform colour distribution and texture characteristics we developed a patch-based method inspired from the well-known image Quilting algorithm by Efros and Freeman [24] widely described in Section 2.4.2. For our purposes we cannot take advantage of the original image Quilting technique as it is. Comparing our context with the authors' results some intuitive disagreements emerge. The first obvious difference we have is that instead of taking blocks just from one (small) input image, we want to build a *tiles dictionary* T collecting vessel-free tiles from the whole healthy subset of the HRF database. This implies that all the patches could be very different from each other so the original error surface e becomes a too weak descriptor of agreement between tiles. We will see in the continuation of this section how we overcame this problem (see page 28). The second main problem is that this technique performs well only in images with uniform illumination, but as we already said this is not our case. So, how can we guarantee a colour intensity distribution consistent with the one in real images? Luckily Efros and Freeman suggest (once again in [24]) a possible solution introducing the concept of *Texture Transfer*. This consists in a little modification of the classic image Quilting technique, basically each tile, to be placed, must also agree with a *Correspondence Map* (CM) that, in our case, should guarantee a *horseshoe*-shaped colour intensity distribution. The background synthesis workflow is visually described by the block diagram in Figure 3.3, use and meaning of the *Cluster Correspondence Map* will be clarified later.

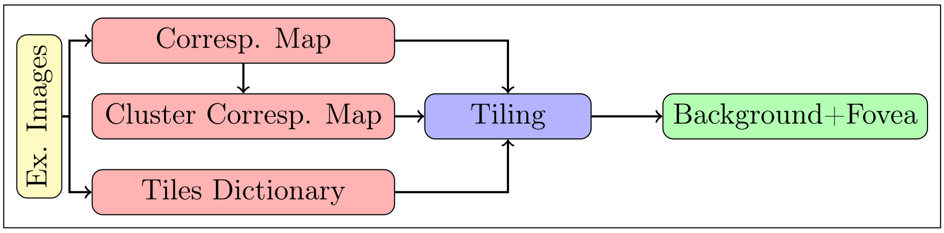


Figure 3.3: The background synthesis block diagram.

Correspondence Map building

For our purposes, a CM is an image that not only recreates a colour intensity distribution consistent with real RFI but also allows a *Quilting-like* tiling algorithm to recreate a realistic texture. As we already mentioned, not all the retinal backgrounds look the same, then we also need a procedure to generate a CM which allows the synthesis of different retinal backgrounds. In order to address those challenges we started again with real RFI. The main idea is to create a CM from (weighted) averaging of original RFI backgrounds. To achieve this goal all the original images have been co-registrated in the same coordinate system. In this step we used a very naive approach; with the VAMPIRE-Annotation Tool we first annotate fovea and OD (centre and radii) of all the images in the HRF healthy subset. Then we translate and rotate them making them share a common polar coordinate system in which the x axis is defined as the line that connects fovea and OD and the y axis obviously is a line perpendicular to that. The origin of this coordinate system is the centre of the OD. In this polar coordinate system the coordinates of each point are defined as: distance from the centre of the Optic Disc (ρ) and angle between the y axis and the point vector (θ). While ρ is the classic Euclidean distance (see Equation 3.1), θ has something different: it is anticlockwise for the right eyes and clockwise for the left ones. This choice keeps the symmetry of the eyes with respect to the nasal and temporal structure. At the end of the co-registration process, all the images share the same x and y axis. The next step is a background estimation. In Section 2.4.1 we presented the idea behind that process which is the use of an image restoration technique to hide the elements of the foreground. Intuitively, at this level, we will consider OD and vessel tree as some generic objects we want to remove. The selected method has been the inpainting technique proposed by Criminisi et al in [21]. With this technique we inpainted the foreground in all the 15 images in the HRF healthy dataset. The target region has been, for each one, its provided vessel GT with a superimposed OD mask obtained by ellipse fitting on five manually selected boundary points (Figure 3.4).

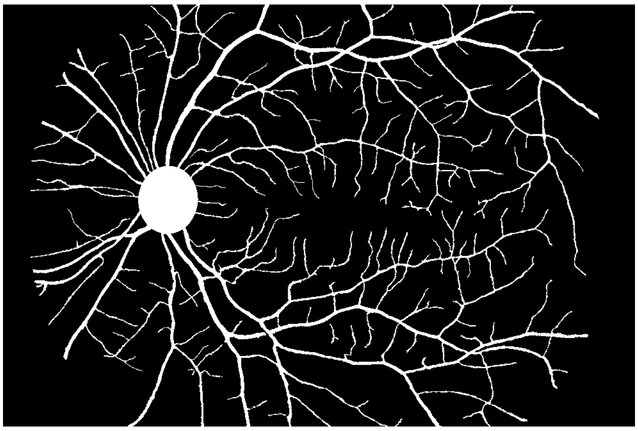
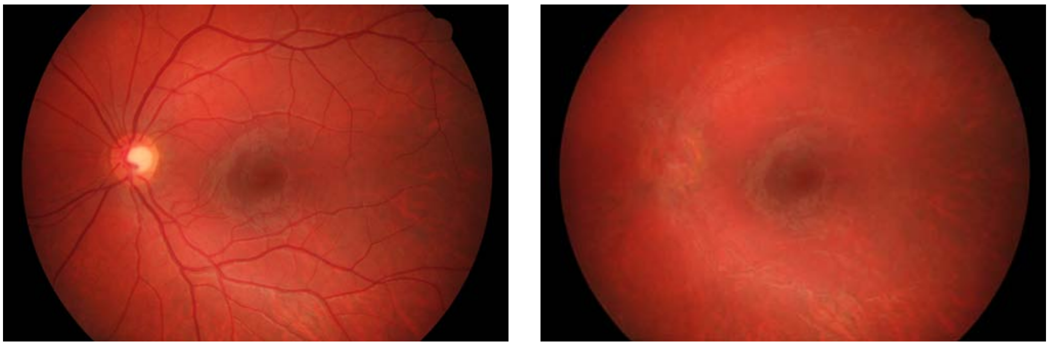


Figure 3.4: A binary mask of a target region.

In the original paper the authors recommend to dimension the template window Ψ in a way that is slightly larger than the largest distinguishable texture element in the source region. Because of the high variability among several retinal backgrounds this parameter is not easily quantifiable. Intuitively, working with high resolution images and small windows takes more time but guarantees better results. We selected square windows 20×20 pixels because this seemed to be a good compromise between elapsed cpu-time and quality of the results. The MATLAB® implementation of Criminisi's algorithm we used performs quite fast on small images, it takes ~ 31 seconds[9] to inpaint a target region that covers 12% of the total area of a 308×206 image using square Ψ sized 4×4 pixel. The original 15 HRF images are 3504×2663 pixels, but, to reduce the computational time before the inpainting, we downscaled them with a resize factor of 0.5. In each image the target region is still approximately the 12%. On the same machine the algorithm runs for almost 36 hours. In Figure 3.5b we can see the obtained result. The vessel tree is no more visible by naked eye but in the larger area of the OD some square shaped artefacts have been created (mainly because of Ψ dimension). However, the result is eligible and it meets our requirements.

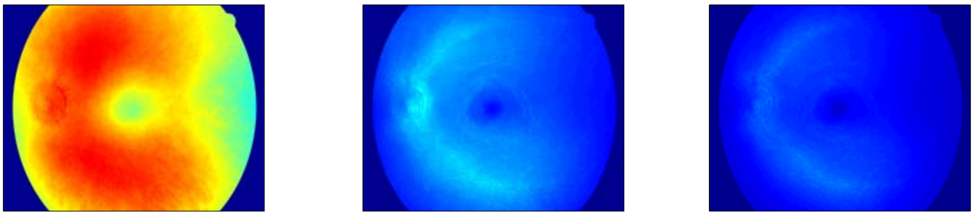


(a) Original RFI. **(b)** Inpainted RFI.

Figure 3.5: A comparison between original and inpainted RFI.

After those two preliminary steps, generating a suitable set of CM becomes finally possible. In order to obtain several appreciably different maps, the following procedure has been iterated. Each CM is obtained by a pixel-wise weighted average of the 15 co-registrated and inpainted original images. In the selection of the weight vector we define three different levels: low (0.1), medium (1.5), and high (3). With 3 levels and 15 images we can generate up to $3^{15} = 14348907$ different CM. Obviously not all of them would be really distinguishable by naked eye, by he way with a wise selection of the weight vector the generation of several different CM is actually possible. For instance to obtain the correspondence colour map depicted in Figure 3.6 a weight vector with high level on the images $\#13-14$, medium on $\#11-12$ and low on the other ones has been used (those numbers are meant as the indexes in the healthy HRF subset). An unoptimised implementation of the CM generation algorithm runs in approximately 40 minutes in full resolution (3504×2336).

[9] On a 2.7Ghz Dual-Core Intel i7.



(a) Red Channel. **(b)** Green Channel. **(c)** Blue Channel.

Figure 3.6: An example of Correspondence Map.

The Tiles Dictionary creation

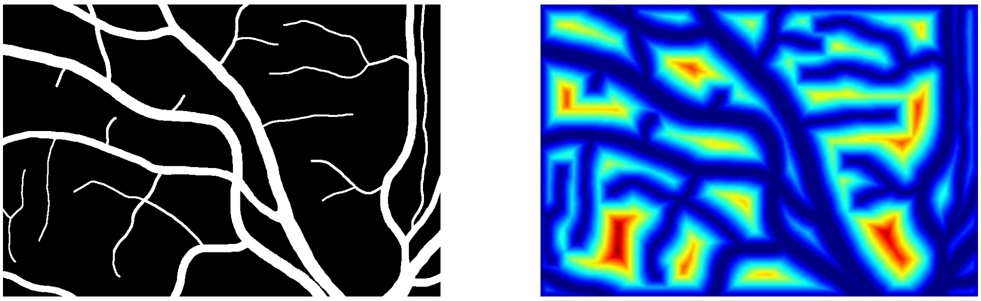
The synthetic background of our RFI phantoms is generated using a patch-based approach. One of the input elements of our algorithm is a collection of blocks (tiles) belonging to real images. The first step is the localisation in the original images of vessel-free areas large enough to allow a patch collection. The second step will be the organisation of those tiles in an appropriate data structure and, last, an accurate analysis of the results. To address the first challenge the information carried by the GT vessel mask has been necessary; the main idea is the use of a pixel-wise distance-from-vessel measure followed by a localisation of the highest values in the obtained distance-map. The two measures taken into account for this application have been the *Euclidean* distance and the *City Block* distance (also known as *Manhattan* distance). In an n-dimensional coordinate system, the Euclidean distance between two points $P(p_1, p_2, \ldots, p_n)$ and $Q(q_1, q_2, \ldots, q_n)$ is defined below.

$$d_E(P, Q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3.1}$$

Picturing it in a 2-D or 3-D space, this intuitively defines the unique shortest-path between two points. This is the standard distance measure, but its application in high resolution images may be too computationally intensive [31]. To overcome this problem we chose to compute an approximation of the actual measure using the below defined Manhattan distance.

$$d_M(P, Q) = \sum_{i=1}^{n} |q_i - p_i| \tag{3.2}$$

Using this measure, the time complexity becomes linear in the number of the pixels of the image and the result is still perfectly suited for our purposes even if it is just an approximation of the real Euclidean distance. We can sketch out the algorithm as follows: let I be a $m \times n$ binary vessel segmentation mask, V its vascular area and $B = I - V$ the non-vascular region (see Figure 3.7a). The maximum possible distance between two points in this scenario can be $\Gamma = m + n$. Let $p_{i,j}$ be the pixel of I in its *i-th* row and *j-th* column, in the initialisation step the algorithm assigns $p_{[i,j] \in V} = 0$ and $p_{[i,j] \in B} = \Gamma$. The final goal is to update all the pixels in B assigning a value corresponding to its distance from the nearest vessel; the value 0 for the



(a) Manual segmentation.

(b) Distance map.

Figure 3.7: A vessel binary mask and its Manhattan distance map.

pixels in V should be intuitively read as *no distance*. The algorithm itself consists in two steps.

1. Raster scan I in top-left → bottom-right direction making the following assignment.

$$p_{i,j} = min\left(\begin{bmatrix} p_{i-1,j-1} & p_{i-1,j} & p_{i-1,j-1} \\ p_{i,j-1} & p_{i,j} & p_{i,j+1} \\ p_{i+1,j-1} & p_{i+1,j} & p_{i+1,j+1} \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}\right) \tag{3.3}$$

2. Repeat step 1 in bottom-right → top-left direction.

To avoid out-of-bound errors the image is initially reframed with a 2-pixel width of Γ-padding. The output of the algorithm is shown in Figure 3.7, the more the pixel is far from the edge of the vessels the more its colour is warm. The algorithm evaluates the distance map in $\sim$ 24 minutes on a 3504×2336 HRF image; this step presents a lot of room for further improvements, in fact many other algorithms which address this challenge with better performances have been proposed [32]. The Manhattan distance map automatically highlights the largest vessel-free areas suitable for a tile collection. For each map in every square shaped area a tile is collected if the side is bigger than 28 pixels; all the data are stored in a preliminary tiles dictionary. At this level each tile in the dictionary has arbitrary size, with the only constraint of being squared with at least 28 pixels side. To allow an easier use of those patches by the proposed Quilting-like algorithm all the tiles are then subsampled in small 7×7 pixels blocks and they are also organised in an appropriate data structure. The size of the tiles has been heuristically chosen, several retinal backgrounds have been synthesised using different tiles of various dimension and the results obtained with 7×7 pixels tiles were the most visually pleasant. The choice of the lower bound for the area (28 pixels) becomes now clear. It has been chosen to have a good compromise between elapsed cpu-time and final dimension of the dictionary. According to that, for being suitable for a collection, a vessel-free area must contain at least 4 tiles. The final tiles dictionary obtained collecting blocks from the usual 15 HRF healthy images has 302683 colour retinal vessel-free tiles.

Tiles Clustering

As we said, our synthesised retinal backgrounds are generated by a Quilting-like tiling algorithm and to faithfully replicate a real colour intensity distribution we used the previously introduced concept of Correspondence Map (see page 21). The tiles dictionary generated from the usual dataset is very large, it contains more than three hundred thousand tiles (7 × 7 pixels each), which means more than fourteen millions of pixels (exactly 14831467). Among those tiles, obviously, some of them are similar in global appearance. From the proposed description of the Quilting algorithm (see Section 2.4.2) it immediately emerges how more likely it is to find a match in similar tiles, instead of in completely random ones. So the idea is to group the tiles in different subsets and look for a possible match only (when possible) between tiles of the same group. Grouping similar objects is a well-known challenge called *Cluster Analysis* and many algorithms have been proposed to address this task. Among those, *K-Means* has been selected because of its simplicity[10]. To consistently link this idea with the CM, a *Correspondence Cluster Map* has been evaluated as well. Briefly, a generic clustering technique tends to create groups (clusters ω_k with $k = 1, \dots, K$) of objects (x_i with $i = 1, \dots, N$) in a large dataset minimising the internal variability and maximising the external one. K-Means tries to do that minimising a cost function which could be, for instance, a measure of distance between each x_i and their group centre $\bar{\mu}(\omega_k)$ also known as *cluster centroid* (see Equation 3.4).

$$\bar{\mu}(\omega_k) = \frac{1}{|\omega_k\ |} \sum_{\bar{x} \in \omega_k} \bar{x} \tag{3.4}$$

In the K-Means algorithm the minimised cost function is the within-cluster *Residual Sum of Squares* (RSS) defined in Equation 3.5.

$$RSS = \sum_{k=1}^{K} \sum_{\bar{x} \in \omega_k} |\bar{x} - \bar{\mu}(\omega_k)|^2 \tag{3.5}$$

An example of its typical output is provided in Figure 3.8; three families of synthetic data extracted from a Gaussian distribution are there correctly grouped into clusters by the K-Means algorithm.

The selection of cluster cardinality is quite easy in case of synthetic data (when it is known a priori), but how can we define an appropriate K in a real application? The hard preliminary selection of K is a well-known drawback of the K-Means algorithm and many methods have been proposed to define it in a robust way [33, 34, 35]. In this work we chose to use the Akaike Information Criterion (AIC) to identify the optimal number of clusters. The general form of AIC is reported in Equation 3.6; $q(k)$ is a penalty index representing the complexity of the tested model with, for instance, k parameters (clusters) and $L(k)$ is the negative maximum Log-Likelihood for the data again with k parameters (clusters).

$$AIC(k) = 2q(k) - 2L(k) \tag{3.6}$$

[10] A detailed description of this method could be found at http://nlp.stanford.edu/IR-book/html/htmledition/k-means-1.html

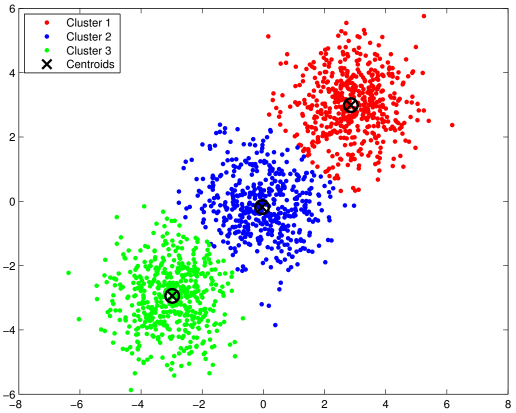


Figure 3.8: K-Means clustering of three Gaussian populations.

In case of K-Means the AIC could be particularised as stated in Equation 3.7; $RSS(k)$ is intended to be as one of the K elements of the external sum of Equation 3.5 and N is the number of elements in the dataset. In a typical fitting problem the term $RSS(k)$ is a goodness-of-fit measure and in this context it should be considered as a measure of distance between the elements x_i and their centroid. It it easy to understand how $RSS(k)$ is monotonically decreasing with increasing K. Particular attention should be paid in case of very large dataset when, because of $N \upuparrows$, it could happen that $2Nk \gg RSS(k)$.

$$AIC_{kmeans}(k) = RSS(k) + 2Nk \tag{3.7}$$

If we run a K-Means on a dataset several times using each time crescent value of k it is possible to plot the trend of $AIC(k)$ Vs the number of clusters; a typical result of this procedure is depicted in Figure 3.9. Intuitively the AIC trend is expected to be monotonically decreasing for low value of k and monotonically increasing when k is high and $2Nk \gg RSS(k)$. To choose the best value for K a possible approach could be the selection of the AIC function minima.

Coming back to our application, we performed a pixel-wise clustering assigning each tile to a single cluster by means of majority vote (every colour tile has $7 \times 7 = 49$ pixels). The selected features for the K-Means clustering have only been the red and green intensities; in retinal fundus images the blue channel is usually more noise affected than the others, so we thought that it could be misleading for the purpose of grouping visually consistent blocks. For the decision of the cluster cardinality let us take into account the AIC trend depicted in Figure 3.9. The generation of this plot has been done running a K-Means with crescent K on 1500 tiles randomly selected from the whole tiles dictionary; according to that the total number of pixel is 73500. We did not use all the dictionary to avoid the problem we might incur with too large dataset ($N \upuparrows$). So, according to the AIC criterion in Figure 3.9 it emerges how the best choice is $K = 4$. Because of the preliminary random selection of initial centroids position in the K-Means algorithm, running again the same experiment the output

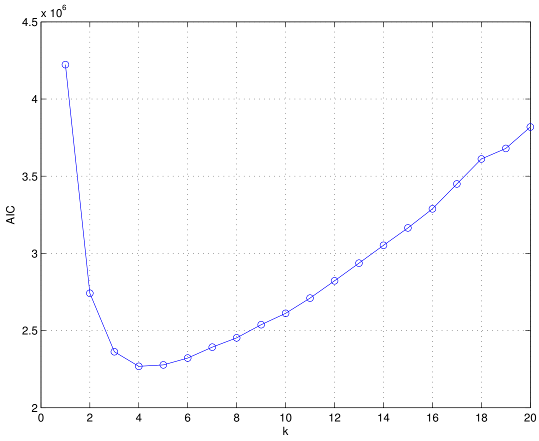


Figure 3.9: AIC Vs Number of Clusters k.

could be slightly different. However, the minima does not change significantly and $K = 4$ is always the best choice. The K-Means result on our pixels is showed in Figure 3.10. The centroid position suggests a higher pixel density near the south edge of the data cloud. This phenomenon is not easily observable in the figure because of the very high number of pixels.

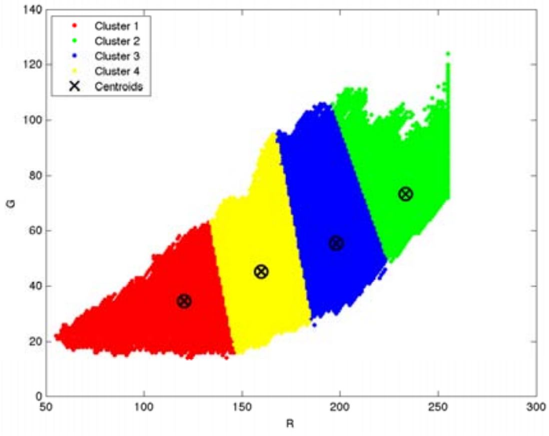
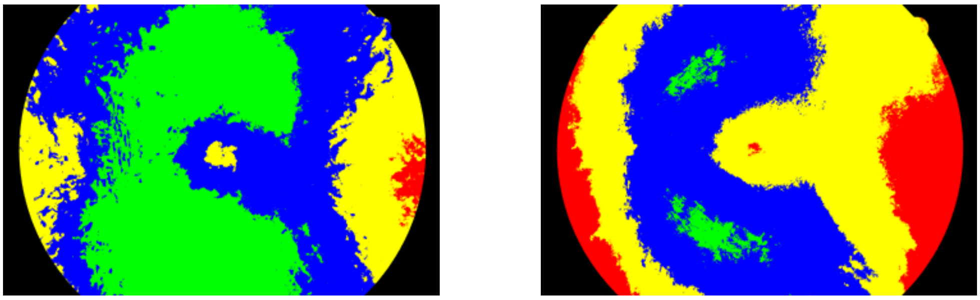


Figure 3.10: Pixel K-Means Clustering.

As previously mentioned, after the assignment of each block to a single cluster, the tiles clustering plays another key role in the background synthesis. Along with the CM a *Cluster Correspondence Map* has been evaluated as well. Its generation follows a very intuitive rule: we evaluate the distance between every pixel of the CM and every cluster centroids, after that we assign each pixel at the nearest cluster. The selected distance measure is the Euclidean's one (see Equation 3.1). Two examples of cluster-CM are showed in Figure 3.11, the colour-code is the same used in Figure 3.10. We can observe how, in the left case (3.11a), a clear majority of the pixels belongs to



(a) Cluster-CM # 1.

(b) Cluster-CM # 2.

Figure 3.11: Cluster Correspondence Maps.

clusters 2, 3 (green, blue) and just few of them are assigned to cluster 1 (red). On the other hand in the cluster map on the right (3.11b) the majority of the pixels are assigned to cluster 4 (yellow) and still a few, but more than before, are assigned to cluster 1. This will allow the tiling algorithm to synthesise two different retinal backgrounds and then to simulate a population of different RFI.

We introduced three main concepts so far, *Tiles Dictionary*, *Correspondence Map* and *Cluster Correspondence Map*. In the next section we will present how the proposed patch-based tiling algorithm generates a convincing retinal background combining together all the three elements.

Proposed tiling algorithm

The original image Quilting technique [24] has already been introduced in Section 2.4.2 (page 13) and, as previously mentioned, this inspired the proposed tiling algorithm. We have already outlined the workflow of the tiling algorithm in Figure 3.3, before going further more into details we can try to better outline it with a practical example. The proposed tiling algorithm reminds us of a jigsaw puzzle solution, those games always consist in a group of coloured pieces (our tiles dictionary) usually sold in a box depicting on top a picture which gives the player a representation of how the solution should look like. In our environment this role is played by the Correspondence Map, which guides the player to correctly place all the pieces (i.e. the tiles). Pretending that the final image is a beach, we could likely have some mainly white/grey puzzle pieces belonging to the sand, some dark blue ones which will represent the sea and some light blue ones for the sky. A possible approach to reach the solution could be making groups of pieces according to their colour and this is exactly the idea that stands behind the K-Means tiles clustering. At this point we must not forget that as it happens in the Quilting algorithm (see Section 2.4.2), to be placed, every single puzzle piece must agree with its neighbours. So far we intuitively explained the meaning of almost every element of our algorithm except the cluster-CM; this matrix could be imagined as the human intuition that can easily associate a target area in the final picture with the previously created group where it is more likely to find a match.

Our method aims at synthesising novel retinal backgrounds stitching together patches collected from many real images and keeping a realistic colour distribution (which is modelled with a CM). This process is visually described by the block diagram in Figure 3.3. At the beginning of this section we sensed that the Quilting algorithm as it is could present some disagreement with our requirements (see page 20). We have already described how we performed the tiles collection, which was the first big difference with the original technique. Let us now define a *new error surface* ε. This measure is obtained from the colour similarity indexes presented by Guobing et al. in 2011 [27] and previously described in Section 2.4.3. Using the same syntax as the original Guobing's paper ε has been defined as stated in Equation 3.8 (where $X \circ Y$ is the element-wise product, also know as *Hadamard product*, between the two matrices X and Y).

$$\varepsilon_{<C_1,C_2>} = 1 - n_{<C_1,C_2>} \circ \gamma_{<C_1,C_2>} \tag{3.8}$$

As the reader remembers, when $C_1 = C_2$ both $n_{<C_1,C_2>} = 1$ and $\gamma_{<C_1,C_2>} = 1$ so $\varepsilon_{<C_1,C_2>} = 0$; the choice of such definition has been made because ε is meant to be an *error* surface, that means a low ε for similar colours and a high ε (with absolute maxima in 1) for different ones.

The algorithm workflow can be described as follows, let Φ be the desired output image, $\tau_k(i)$ the *i-th* element of the tiles dictionary T assigned to the cluster ω_k, let also σ_{gen} be a generic threshold. The final result can be achieved iterating the following steps.

- Go through Φ in raster scan order in steps of the size (l) of the tiles minus the size (b^{ov}) of the overlap; the default values for them are $l = 7$ and $b^{ov} = \lceil \frac{1}{3} \times l \rceil = 3$.

- For every location in the CM collect the (r, c)-centred tile $\hat{\tau}(r, c)$ and get the correspondent cluster label k, which is defined as the value that appears more often (i.e. the mode) in the $l \times l$ matrix centred in (r, c) in the cluster-CM.

- Continue to randomly pick up a $\tau_k(i)$ from ω_k until it satisfies all the constraints below.

    - Consistence with $\hat{\tau}(r, c)$, which means $d(\tau_k(i), \hat{\tau}(r, c)) < \sigma_1$ where the d operator denotes the Euclidean distance (see Equation 3.1).
    - Consistence between the overlap areas B_1^{ov} and B_2^{ov}, which means more than the 90% of the pixels of the error surface $\varepsilon_{<B_1^{ov},B_2^{ov}>}$ must be minor than σ_2 (in case of multiple overlaps this step has to be repeated in both the *North* and the*West* side).

- Once the best match is selected, we compute the *Cumulative Minimum Error* between the overlapping areas defined as $E_{i,j} = \varepsilon_{i,j} + min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$ where $i = 2, \ldots, N$ $j = 2, \ldots, M$ and ε is $N \times M$.

- Evaluate the Best Boundary Cut (BBC) scanning row-wise E and selecting for each row its minimum value.

- Place finally the new tile according to the BBC and repeat.

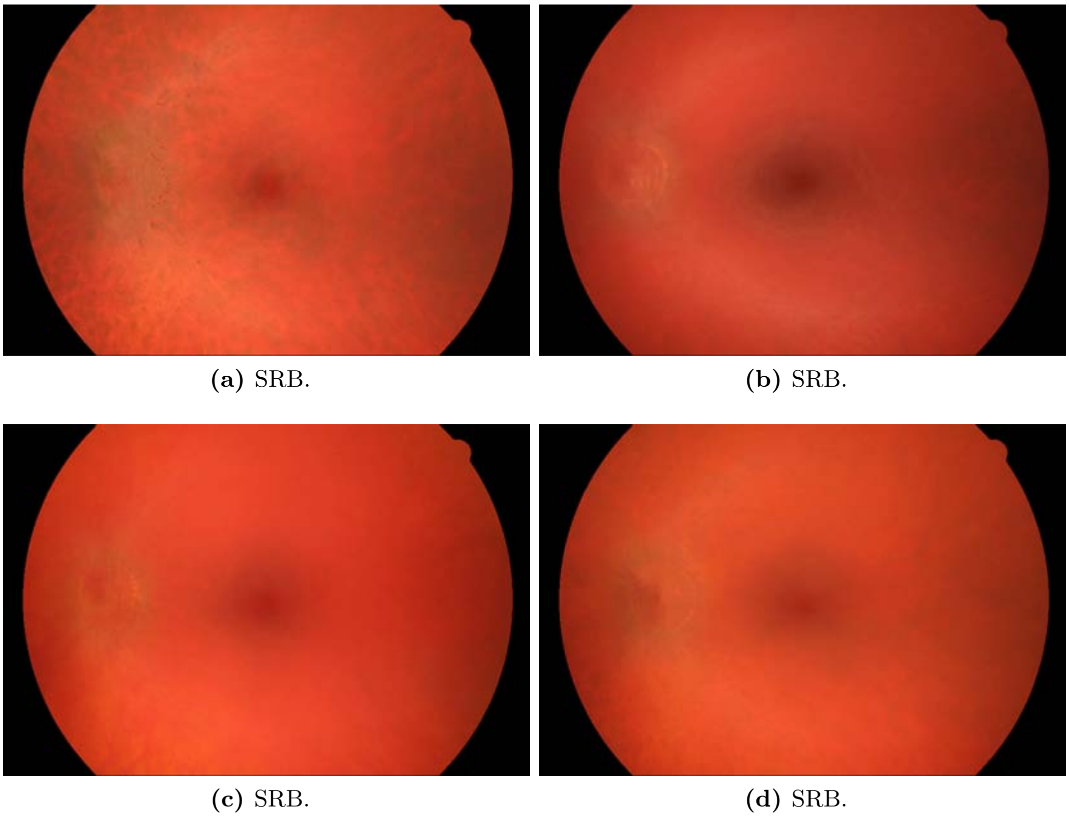
The two thresholds σ_1 and σ_2 have been heuristically determined. The first one intuitively determines how much the output image should be influenced by the CM and the second one determines the level of agreement among neighbouring tiles. There is still a lot of room for further improvements, for example the definition of a protocol for robust dimensioning of those two thresholds can improve not only the quality of the output but even the elapsed cpu-time. However, our preliminary results look promising (see Figure 3.12); each generated retinal background seems to reproduce a realistic colour intensity distribution and no raw tiles are recognisable by naked eye. Comparing Figure 3.12a and Figure 3.12b we can see how with the proposed technique a realistic representation of choroidal vessels can be achieved and, as in real cases, their contribute could heavily influence the whole retinal background or could just be a local effect. Another advantage of our technique is that it automatically provides a realistic representation of the fovea. The four results presented in Figure 3.12, in fact, clearly show a darker region in their centre which really looks like an original fovea. The first drawback that may emerge from those results is the presence of some square shaped artefacts at the left of each fovea, in a region that our intuition could associate with the OD area. In order to avoid those artefacts, a reduction in the dimension of the template window at the inpainting step is needed. By the way reducing its size means dramatically to increase the elapsed cpu-time. We have already discussed our choice in Section 3.3.1 and we previously pointed out how the presence of some artefacts in this area is trivial because they are going to be totally hidden by a synthetic superimposed OD (see Section 3.3.2). Figure 3.12 shows four examples of left-eye retinal backgrounds, but our algorithm could obviously synthesise right ones as well.

In this section we provided a detailed explanation of all the steps which are necessary to generate synthetic retinal backgrounds and foveae, this must be the first step for the generation of synthetic RFI. Our work aims at generating non-vascular regions so in the next section we will present the proposed Optic Disc synthesis technique.

3.3.2 Optic Disc generation

The parametric model formulation

For the Optic Disc synthesis a completely different approach has been developed. Its generation, in fact, has not been patch-based (as the background and fovea one) but model-based. The idea is to learn the distribution of key morphometric quantities of the Optic Disc from real images. This would require a large number of retinal fundus images in which the target parameters have been estimated. In this short project, we could only estimate the parameters from 30 (15 healthy and 15 with DR) images; time did not allow measuring larger sets of images (which would have been available via the VAMPIRE software suite and also public repositories), and we could not locate studies reporting normative values on large populations for the proposed parametric model. Results suggest however that the algorithms we propose support the generation of realistic Optic Discs even with modest amounts of training data. As already done for background and fovea, we started the development of synthetic OD from the observation of real ones (Figure 3.14a and 3.14e). The OD generation workflow is outlined by the block diagram in Figure 3.13.



(a) SRB.

(b) SRB.

(c) SRB.

(d) SRB.

Figure 3.12: Synthetic Retinal Backgrounds.

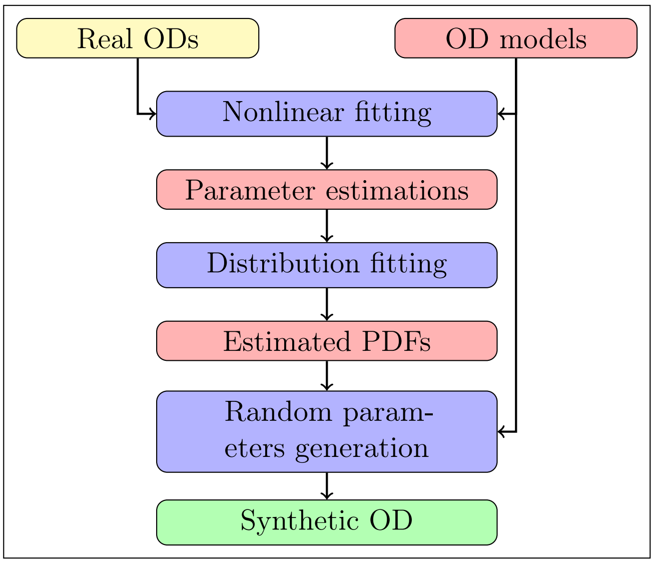


Figure 3.13: The OD generation block diagram.

OD texture and appearance has already been described in Section 2.3.2 where the idea of using a 3-D surface for its synthesis, in a (r, g, b) colour space, has been provided as well. In order to define a suitable parametric surface an accurate observation of a large number of real ones is mandatory. Figure 3.14 shows us two examples of ODs and their 3-D colour intensity surfaces; in each graph the horizontal plane corresponds to the image (i, j) dimensions and the vertical axis is one of the red, green or blue intensity in the correspondent channel. Trying to figure out which parametric model could fit those data, an immediate evidence emerges: the red channel has a very different shape with respect to the other two. With a close observation of Figure 3.14b and 3.14f it is possible to see how in red channel the blood vessels are not really influent and the global shape recalls a cylindric *plateau* with an elliptic upper base. From our physiological knowledge about the OD we know that it usually appears as an ellipse with a small inner depression called cup. Watching Figure 3.14b and 3.14f it may seem that in the red channel the external OD boundaries are better highlighted while the cup is more visible in green and blue channels (see Figures 3.14c 3.14d 3.14g 3.14h). In this regard we may also remember that vessel detection algorithms usually work just with the green channel because it shows the best vessel/background contrast [4, 3] while, on the other hand, in OD detection techniques the information carried by the red channel is taken into account. To our best knowledge those algorithms generally work with images converted in greyscale [30, 20]. Looking at green channel (Figures 3.14c 3.14g) the first difference with respect to the red one that immediately emerges is the completely different shape, the upper part of the surface is not flat but it looks more like a bell. Trying to eliminate by visual interpolation the strong contribute of the vessels, we can observe a change of the trend approximately in the middle of the surface. It looks as if the whole green channel shape can be modelled again with a plateau but with a bell-like shaped hat on top. Again about the green channel comparing Figure 3.14g and Figure 3.14h we can see how the same vessel is easier to see in the green than in the blue channel; unlike most common situations for our purposes the blood vessels are considered as noise. We will see in the next section how their contribute has been skipped avoiding the use of preliminary image filtering. The last observation we can make is about the blue channel, from the surfaces in Figure 3.14d and 3.14h we can conclude that the blue channel has roughly the same shape as the green one, even if the trend change in the middle of the body is less stressed. So the next step will be the formulation of a three dimensional plateau surface with and without a bell-like shaped hat.

In order to achieve our goals, we first started with the study of a suitable 2-D sigmoid signal that could be, as a second step, rotated along a vertical axis in order to generate a 3-D solid of revolution. The selected curve has been the *Logistic Function* (see Equation 3.9).

$$f(x) = \frac{1}{a + be^{-cx}} \tag{3.9}$$

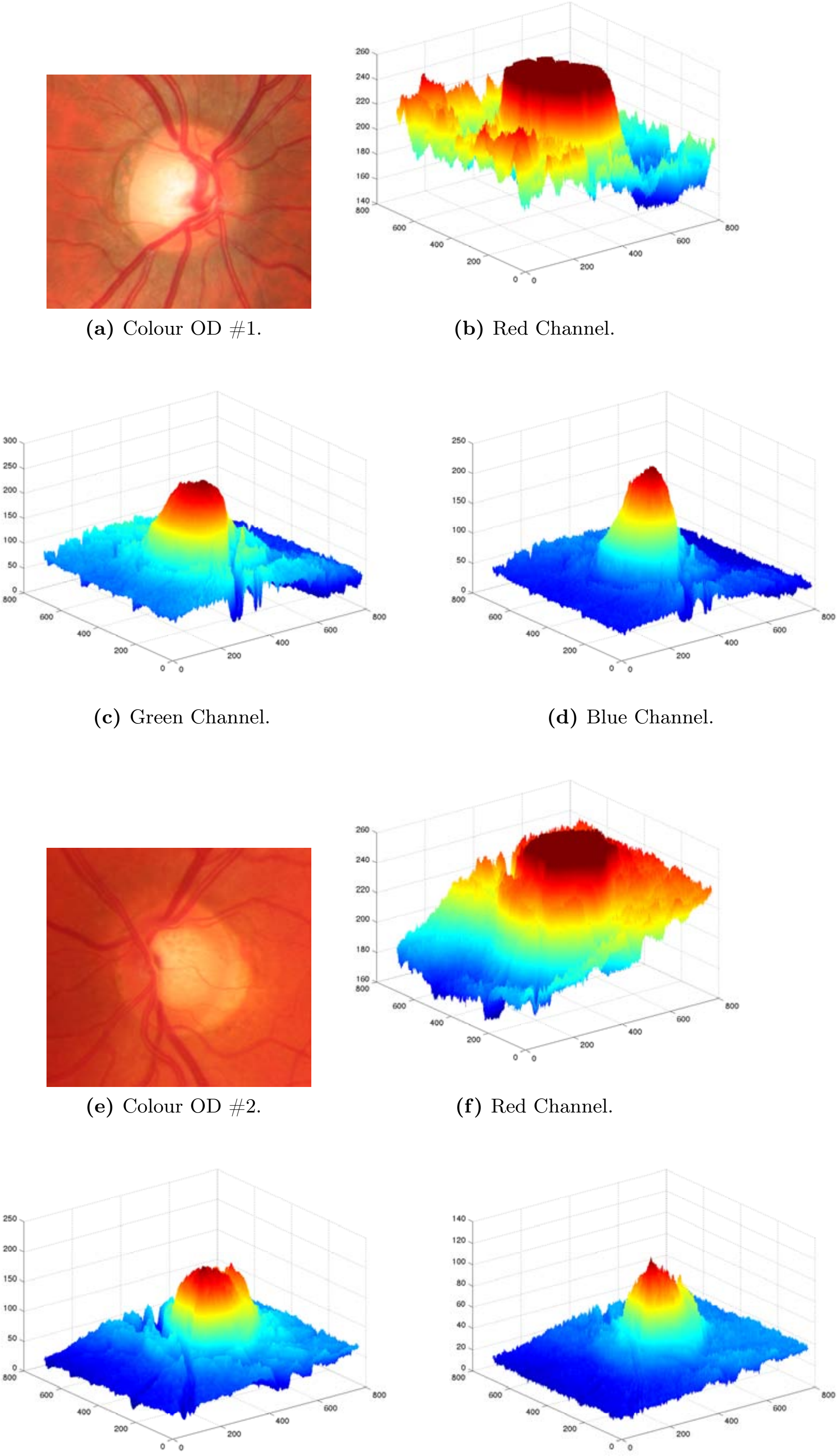
We may sense that a variation of the parameters $[a, b, c]$ can dramatically influence the trend of the curve. In Figure 3.15a, some Logistic Function with $b = c = 1$ and $a \in [1, \cdots, 9]$ is showed; we can observe how with an increase of a (in the denominator in Equation 3.9) the curve tends to reduce its amplitude. In Figure 3.15b some plots with $a = c = 1$ and $b \in [1, \cdots, 9]$ are represented. We can see how increasing the b value the curve tends to meet the vertical axis in a lower point without changing the amplitude. Finally in Figure 3.15c some representations of the Logistic Function with $a = b = 1$ and $c \in [1, \cdots, 9]$ is shown. Tuning c allows to modify the stiffness without changing the amplitude or the intersection with the vertical axis. Tuning just three parameters the Logistic Function can be easily modulated and can assume infinite different shapes. On the same line we can formulate a 3-D Logistic Function provided with a number of tuneable parameters.

A first definition of the plateau surface equation (i.e. the red channel colour intensity model) is stated in Equation 3.10 (where $\bar{p}_1 = [x_0, y_0, z_0, a, \sigma_1, \sigma_2]^T$ is the parameter vector).

$$f_1(x, y, \bar{p}_1) = z_0 - \frac{1}{a + \exp[-(\frac{x-x_0}{\sigma_x})^2 - (\frac{y-y_0}{\sigma_y})^2]} \tag{3.10}$$

As previously done for the Logistic Function, we can provide an intuitive explanation of the role played by every parameter of the vector $\bar{p}_1$. The first three (x_0, y_0, z_0) simply control the translation of the surface along the three x, y, z axes. More sophisticated meaning has a, in fact, as its homologous in Equation 3.9, it controls the amplitude of the surface. We may observe that $\lim_{a\to\infty} f_1(x, y, \bar{p}_1) = z_0$ while for $a \to 0$ the surface $f_1(x, y, \bar{p}_1)$ tends to be the additive inverse reciprocal of a two-dimensional Gaussian function (with a z_0 offset on the vertical axis). The two parameters σ_1 and σ_2 play intuitively the same role as the standard deviation parameter (SD) in a Gaussian function. They control the spread of the two independent variables x and y along the respective axis. Using the parameters values specified in Table 3.1 Equation 3.10 generates the surface depicted in Figure 3.16a.

The first observation we can make about this surface is that changing the value of σ_1 and σ_2 we can simulate plateaux just with circular or elliptical top base, this does not seem very realistic. Looking at many real OD, in fact, it is sometimes possible



(a) Colour OD #1.

(b) Red Channel.

(c) Green Channel.

(d) Blue Channel.

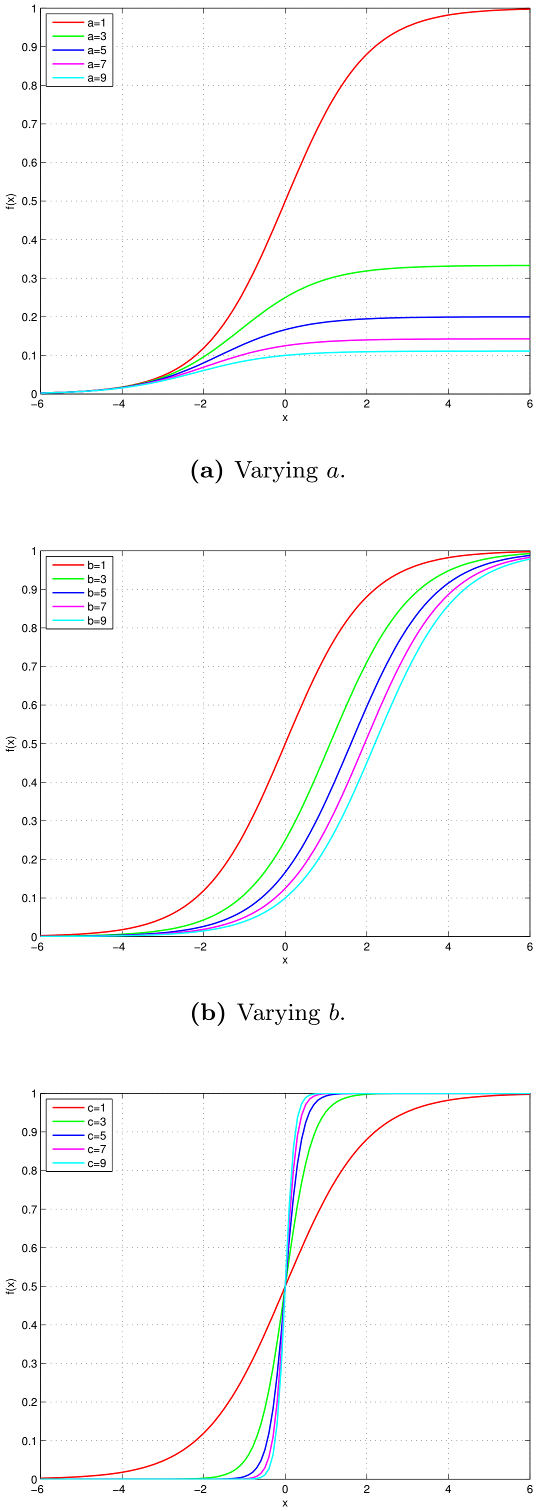
(e) Colour OD #2.

(f) Red Channel.

(g) Green Channel.

(h) Blue Channel.

Figure 3.14: Real ODs and their colour surfaces.



(a) Varying a.

(b) Varying b.

(c) Varying c.

Figure 3.15: The Logistic Function with varying parameters.

x_0	250
y_0	250
z_0	250
a	0.015
σ_1	100
σ_2	100

Table 3.1: Parameters of the surface in Figure 3.16a.

x_0	250
y_0	250
z_0	250
a	0.015
σ_1	100
σ_2	100
A	8
ω	4

Table 3.2: Parameters of the surface in Figure 3.16b.

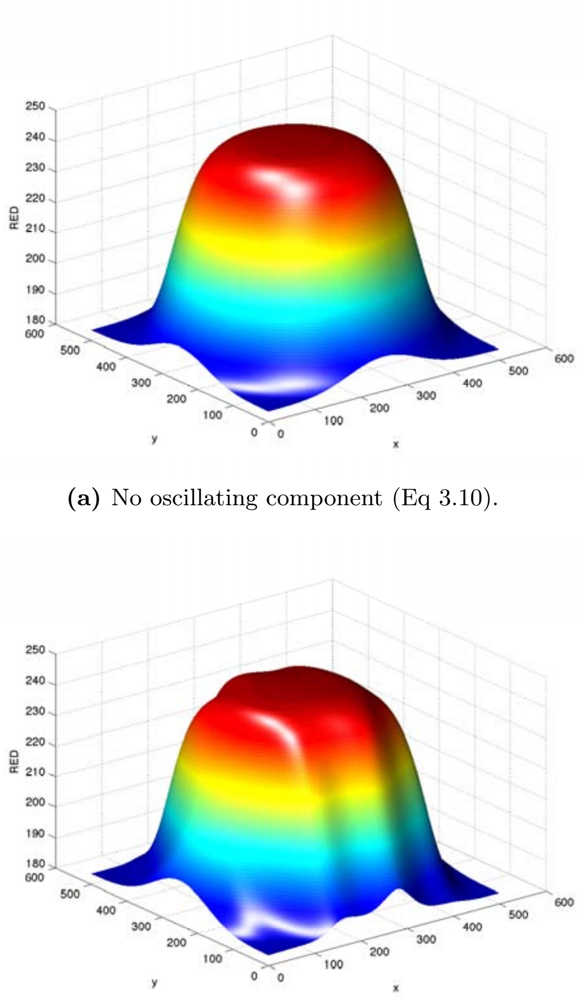
to understand how the contour of their edge can be slightly irregular. We thought to model this phenomenon introducing an oscillating component with very low angular frequency, so $\bar{p}_1$ becomes $\bar{p}_2 = [x_0\ y_0\ z_0\ a\ \sigma_1\ \sigma_2\ A\ \omega]^T$ and Equation 3.10 becomes Equation 3.11 (where $t \in [0, \cdots, 2\pi]$).

$$f_2(x, y, \bar{p}_2) = z_0 - \frac{1}{a + \exp[-\left(\frac{x-x_0+A\cos(\omega t)}{\sigma_x}\right)^2 - \left(\frac{y-y_0+A\cos(\omega t)}{\sigma_y}\right)^2]} \tag{3.11}$$

Equation 3.11 generates the surface in Figure 3.16b using the parameters in Table 3.2. The new formulation of the model has 7 parameters, the nonlinear least square fitting algorithm we used for their estimation did not give a reasonable uncertainty for all of them (see next section), so a new formulation of the model with few parameters has been needed. The first idea we had was to merge the two SD-like parameters, in a way that $\sigma_1 = \sigma_2 = \sigma$. This would not allow to have a different spread of x and y along the respective axis and that might generate an always circular shaped top base. But actually the contribute of the two introduced oscillating elements already gave the final surface a more realistic irregular and elliptic-like top base (see Figure 3.16b where the oscillating terms have been stressed on purpose).

After those observations we can finally define $f_2(x, y, \bar{p}_2)$ as the model we used for the synthesis of our OD red channel where ($\bar{p}_2 = [x_0, y_0, z_0, a, k, \sigma, \sigma, A, \omega]^T$), from now on we will refer to that as $f_R(x, y, \bar{p})$.

The next step must be the definition of another surface for the synthesis of both the green and the blue channels. We already observed how their shape needs a different parametrical model with respect to the red one. Empirical observations



(a) No oscillating component (Eq 3.10).

(b) Oscillating component (Eq 3.11).

Figure 3.16: The Plateau surface.

x_0	250
y_0	250
z_0	250
a	0.015
σ	100
A	8
ω	3
x_0^*	3
y_0^*	3
k	50
σ^*	60

Table 3.3: Parameters of the surface in Figure 3.17.

of Figure 3.14 led us to the conclusion of the necessity of a plateau function with a bell-like hat. The idea is to use a new model made of a sum of two components: $f_R(x, y, \bar{p})$ and a simple Gaussian surface $G(x, y, \bar{q})$ which models the inner cup colour intensity. The final surface formulation is stated in Equation 3.12 where $\bar{q} = [x_0^* \; y_0^* \; k \; \sigma^*]$.

$$f_{GB}(x, y, \bar{q}) = f_R(x, y, \bar{p}) + \underbrace{k \exp\left[-\left(\frac{x - x_0^*}{\sigma*}\right)^2 - \left(\frac{y - y_0^*}{\sigma*}\right)^2 \right]}_{\text{G(x,y,}\bar{q}\text{)}} \tag{3.12}$$

The model stated in Equation 3.12 with the parameters in Table 3.3 generates the surface in Figure 3.17. The first observation we can make is that the two components of f_{GB} must potentially have a different offset along the x and y dimensions. In fact from our physiological knowledge we know that the centre of the inner cup is usually placed closer to the external OD edge in its fovea side and it does not coincide with the global OD centre. This visual observation is mathematically modelled with the introduction of two new parameters x_0^* and y_0^* in $G(x, y, \bar{q})$ which allow the cup centre to be slightly translated with respect to the global OD one. The second observation we can make is that the SD term along the x and y axes are $\sigma_x = \frac{\sigma^*}{\sqrt{2}} = \sigma_y$, which means that the spread of the Gaussian term along the two components in the horizontal plane is always the same and the inner cup is circular. This loss of degree of freedom is obviously a simplification and the solution with two different terms has been explored as well. Results showed us how the introduction of another parameter did not significantly improve the goodness-of-fit and the difference in the rendered OD was not visually appreciable. So we decided to use this simplification pairing the two SD terms in the Gaussian function.

Parameters estimation

The proposed models for the synthesis of ODs in the (r, g, b) colour-space have just been described (see Equations 3.11 and 3.12) and in this section we will define the followed parameters estimation procedure. As a second step we will aim at

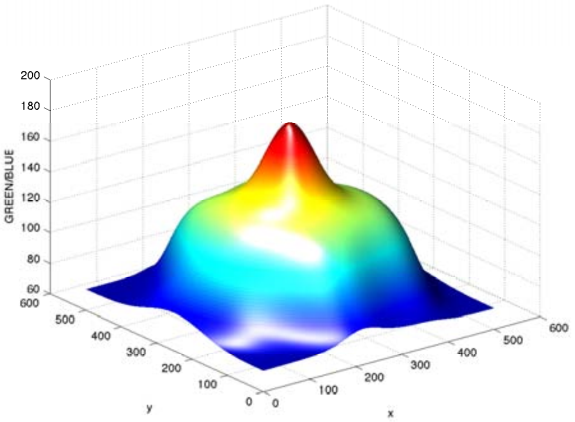


Figure 3.17: Green and blue channel surface.

inferencing a suitable Probability Density Function (PDF) for every parameter of the proposed models (see next section). In order to do that the usual database consisting in just 15 healthy HRF retinal images seemed not to fulfil our requirements. So we decided to extend the input data with the information carried by other 15 images belonging to the diabetic HRF subset. The final number of 30 input images barely seemed to be enough for the estimation of our parameters with a reasonable uncertainty level and, consequently, for the inference of a suitable PDF for each one of them. As the reader remembers, the HRF database consists in three subset: healthy, diabetic and glaucomatous retinal images. This work aims at generating synthetic retinal fundus images recognisable as healthy, so only the healthy subset has been taken into account so far. To our best knowledge, in RFI collected from patients affected by DR some structures, called hard exudates, are clearly recognisable in the background but the OD is not significantly modified. The glaucoma, instead, mainly affects the OD itself and modifies the CDR. Estimating a set of parameters for the synthesis of healthy OD was our purpose and because of that we extended our database with the DR subset. A future step could be the characterisation of the presented method in case of various kinds of diseases. Potentially our technique could simulate the presence of glaucomatous ODs, hard exudates, drusens, retinal lesions and so on. Time did not allow us to develop those features, but the presented work is, indeed, the right starting point for future improvements.

Coming back to our technique, in order to estimate the parameters of the models we took advantage of the *Nonlinear Least Squares* (NLLS) criterion implementation provided in the custom MATLAB® function `lsqnonlin`[11]. As already mentioned in the previous section, an occurred non-trivial problem has been dealing with the strong contribute carried by the blood vessels which, for our purposes, can be assumed as noise. We wanted to overcome this issue avoiding any modification of the input data. As previously done in the background generation procedure we could have thought to de-noise the OD using, for instance, inpainting techniques. This should have implied the introduction of some blur or artefacts depending on the selected inpainting technique (see Section 2.4.1 on page 11) decreasing the reliability of both input data and, consequently, parameter estimations. So, instead of modifying the input data, we simply decided to choose a weighted NLLS criterion assigning zero weight to the vascular regions identified by the manual vessel segmentations. This criterion evaluates the parameters estimation ($\hat{p}_{WLS}$) attempting to iteratively solve the problem stated in compact matrix-vector in Equation 3.13 (MATLAB® uses the trust-region-reflective algorithm by default), where p and z are, respectively, the parameter and the input data vectors, $G(p)$ is a generic nonlinear model and Σ_w is the diagonal weight matrix.

$$\hat{p}_{WLS} = \arg\min_{p} ||z - G(p)||^2_{\Sigma_w^{-1}} \tag{3.13}$$

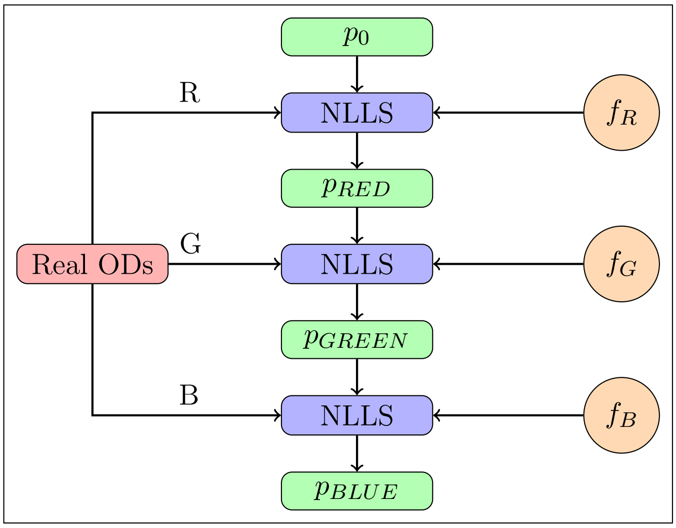


Figure 3.18: The parameters estimation process.

In our case z and $G(p)$ have respectively been, each time, the vectorisation of one of the three channels of a square-shaped region with side 700 pixels containing the real OD (i.e. the real data) and the consistent colour channel model. In order to produce reliable results, every minimisation algorithm, needs to be initialised with an n-dimensional starting point (also known as seed point, p_0), where n is the number of parameters. The value of n itself must be carefully dimensioned. We will now present two tricks we made which address those two problems. In the proposed procedure, for each image, we first estimated the parameters of the red channel then the green and finally the blue one. A first p_0 is heuristically determined equal for all the red channels; after that, for the common parameters between f_R and f_{GB}, the NLLS algorithm uses as seed point the results of the previous step. In other words, the parameters estimation procedure for the red (green) channel produces, at the same time, both its results and the seed point for the green (blue) one. This process is visually described in Figure 3.18. The choice of using the results of the previous step as seed point for the next one consistently reduced the parameter estimation uncertainty. In the passage between red and green channel, for the parameters belonging to the Gaussian term in Equation 3.12 ($\bar{q}$), this trick obviously was not

[11] Please refer to the MATLAB® Help for further details.

applicable. So a manual tuning of those seed points has been provided. The second observation we made was to accurately dimension the number of the parameters. We may now observe that the two models $f_R(x, y, \bar{p})$ and $f_{GB}(x, y, \bar{q})$ have 7 and 11 parameters each. Investigating on the intuitive meaning of those parameters, we provided simplifications. A further observation we can make is that in Equation 3.12 the first component of the green and blue channel model has exactly the same formulation as the red one. We previously observed how the red channel gives the OD its external boundaries while the green and blue channels mainly model the inner cup. Because of that, during the estimation of the parameters for the second and third channel, a new quantification of parameters like x_0, y_0, σ, A and ω may be redundant. In fact, as the reader remembers, those parameters identify the position of the OD centre, its spread along the x and y coordinates (approximately its radius) and influence its external shape. The low frequency oscillating component models its irregular edges and it is very unlikely that those quantities could be different from one channel to another. So during the parameters estimation step for the second and the third channel the parameters above are not quantified again and the same value estimated in the first step has been imposed. This trick let us decrease the number of parameters for f_{GB} from 11 to just 6.

The previously mentioned concept of uncertainty in the parameters estimation process is still not formally defined. In order to understand its formulation, let us characterise Equation 3.13 in a simple linear case. The problem can be formulated in the compact matrix-vector notation as $z = G \cdot p + v$, where, as usually, z and p are the input data and the parameters vector respectively, v is an additive white Gaussian noise (AWGN) vector and G can be formulated as stated in Equation 3.14.

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & & \\ g_{n1} & g_{n2} & \cdots & g_{nm} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{3.14}$$

Obtaining p from Equation 3.14 is a closed-form problem and its solution assumes the well-known formulation defined below (Equation 3.15).

$$\hat{p}_{WLS} = (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} z \tag{3.15}$$

Where Σ_w is a generic diagonal weight matrix (e.g. it could be the AWGN covariance matrix Σ_v). In this context, the estimation error $\tilde{p}$ can be defined as $\tilde{p} = p - \hat{p}$ where p is the *true* value of the parameters (i.e. ideally not noise affected) and $\hat{p}$ is the *estimated* one. In this case, considering the general problem definition $z = G \cdot p + v$ and looking at Equation 3.15, $\tilde{p}$ can be reformulated as follows.

$$\begin{gathered} \tilde{p} = p - \hat{p} = p - (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} z = \\ p - (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} G p + (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} v = \\ [I_m - (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} G] p + (G^T \Sigma_w^{-1} G)^{-1} G^T \Sigma_w^{-1} v = Dp + R \end{gathered}$$

Where D is a deterministic term while R is a random one. This implies that the covariance matrix of $\tilde{p}$ can be easily defined as covariance of just the random term R and $\Sigma_{\tilde{p}} = (G^T \Sigma_w^{-1} G)^{-1}$ is a square $m \times m$ matrix as stated below (Equation 3.16).

$$\Sigma_{\tilde{p}} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2m}^2 \\ \cdots & \cdots & \ddots & \cdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \cdots & \sigma_{mm}^2 \end{bmatrix} \tag{3.16}$$

The diagonal terms σ_{ii}^2 are the standard deviation of the estimation of the *i-th* parameter $\hat{p}_i$. From now on, in the presented work we refer to the estimation uncertainty in the sense of percentage *Coefficient of Variation* (CV), which is defined as follows (Equation 3.17).

$$cv(\hat{p}_i) = 100\frac{\sigma_{ii}}{\hat{p}_i} \tag{3.17}$$

Coming now back to our nonlinear problem case, Equation 3.13 now does not have unique solution and $\Sigma_{\tilde{p}}$ can only be approximated as stated in Equation 3.18.

$$\Sigma_{\tilde{p}} \approx (S^T \Sigma_w^{-1} S)^{-1} \tag{3.18}$$

Defining the residual vector $R(\hat{p}) = z - G(\hat{p})$, the term S can be defined as follows (Equation 3.19).

$$S = \begin{bmatrix} \left.\frac{\partial R_1}{\partial p_1}\right|_{p=\hat{p}} & \left.\frac{\partial R_1}{\partial p_2}\right|_{p=\hat{p}} & \cdots & \left.\frac{\partial R_1}{\partial p_m}\right|_{p=\hat{p}} \\ \left.\frac{\partial R_2}{\partial p_1}\right|_{p=\hat{p}} & \left.\frac{\partial R_2}{\partial p_2}\right|_{p=\hat{p}} & \cdots & \left.\frac{\partial R_2}{\partial p_m}\right|_{p=\hat{p}} \\ \cdots & \cdots & \ddots & \cdots \\ \left.\frac{\partial R_n}{\partial p_1}\right|_{p=\hat{p}} & \left.\frac{\partial R_n}{\partial p_2}\right|_{p=\hat{p}} & \cdots & \left.\frac{\partial R_n}{\partial p_m}\right|_{p=\hat{p}} \end{bmatrix} \tag{3.19}$$

The matrix in Equation 3.19 could be obtained from the `lsqnonlin` MATLAB® function which could be called with a command like the following one.

```
[p,resnorm,residual,exitflag,output,lambda,jacobian] = lsqnonlin(@myfun,p0)
```

Here `fun` is a MATLAB® function which defines a nonlinear model, such as $G(p)$, and returns in `F` a cost function (e.g. the weighted residuals).

```
function F = fun(p)
F = ...             % Compute function values at p
```

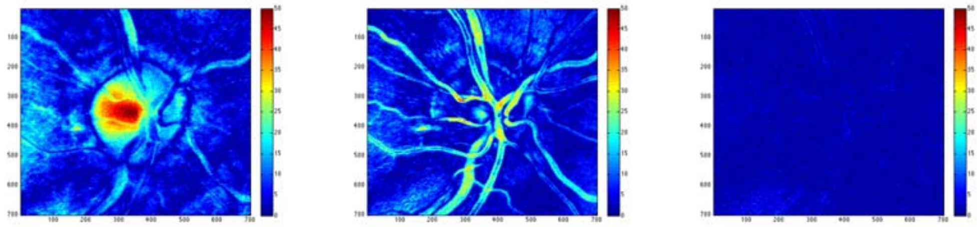
The first output argument of `lsqnonlin` is the vector of the parameters giving the best fit. Skipping the next five ones, which are not of our interest, we now focus on the last output argument. This defines the matrix `jacobian` which, according to the MATLAB® Help, is defined as the Jacobian (J) of `fun` at the solution `p`. With the syntax introduced above, the solution `p` is our $\hat{p}$ and the covariance matrix of the estimation error $\tilde{p}$ of Equation 3.18 can be evaluated as $\Sigma_{\tilde{p}} = (J^T J)^{-1}$. So, with the same idea already outlined for the linear case, we can approximate the SD of the *i-th* parameter $\hat{p}_i$ as the square root of the elements on the main diagonal of $\Sigma_{\tilde{p}}$ and, consequently, evaluate their uncertainty applying Equation 3.17. Table 3.4 summarises the parameters estimation average values (AV) and uncertainty (CV).

Channels	Red		Green		Blue	
Parameters	AV	CV%	AV	CV%	AV	CV%
x_0	341	8.925	∅	∅	∅	∅
y_0	323	1.054	∅	∅	∅	∅
z_0	235	0.154	98	1.780	51.696	3.020
a	0.019	0.965	0.049	7.948	0.117	52.178
σ	152.963	1.003	∅	∅	∅	∅
A	23.885	53.681	∅	∅	∅	∅
ω	1.6365	43.059	∅	∅	∅	∅
x_0^*	∅	∅	288.366	4.224	321.673	1.778
y_0^*	∅	∅	313.947	5.304	336.904	2.510
k	∅	∅	69.054	32.423	45.759	14.985
σ^*	∅	∅	127.160	168.216	181.322	11.578

Table 3.4: Parameters estimation values and uncertainty.

As we can see, except for few parameters, the average uncertainty always has a reasonable value. We must specify that, in extreme cases like σ^* in the green channel, there is just 1 image (out of 30) where the parameter estimation fails (CV% $\rightarrow \infty$) which drives the average value to a unreasonable high value. So, for this parameter, and for all of them which have CV% major of a certain threshold (e.g. 10%), the images where the parameters estimation failed have been excluded from further analysis. The number of excluded images has never been bigger than 3 and this does not heavily influence the reliability of the next analysis, even in our data-poor situation. The symbol ∅ has been placed in those cells which have no meaning, for instance, the value of x_0 and y_0 which denotes the position of the OD centre in the squared 700×700 window, has not been estimated in the green and blue channels.

The next mandatory step is the discussion about the fitting performances. Figure 3.19 shows an example of absolute residuals (the image #10 in the DR subset). As we can see, the residuals on the red channel (Figure 3.19a) are correctly higher on the inner cup, which is not modelled by f_R, and lower near the OD external boundaries. On the other hand figure 3.19b shows how the blood vessels, which show the best contrast on the green channel, have not been taken into account by our weighted NLLS criterion. They show higher absolute residual value because of the zero weight we gave them during the parameters estimation procedure. We may also observe that the model seems to correctly fit the real OD green intensity in its brighter inner cup. Residuals in the blue channel (Figure 3.19c) have also been reported, but the high level of noise did not allow any observation there. Table 3.5 summarises the fitting performances, in terms of mean absolute error (MAE) specific for each channel, for all the 30 images of the selected dataset. Let Z be a 700×700 matrix containing the real OD colour intensity for one channel and $G(\hat{p})$ the correspondent fitting surface, their MAE is defined as stated in Equation 3.20.



(a) Red channel. **(b)** Green channel. **(c)** Blue channel.

Figure 3.19: Example of absolute residuals.

$$MAE = \frac{1}{1400} \sum_{i=1}^{700} \sum_{j=1}^{700} |Z(i,j) - G(i,j,\hat{p})| \tag{3.20}$$

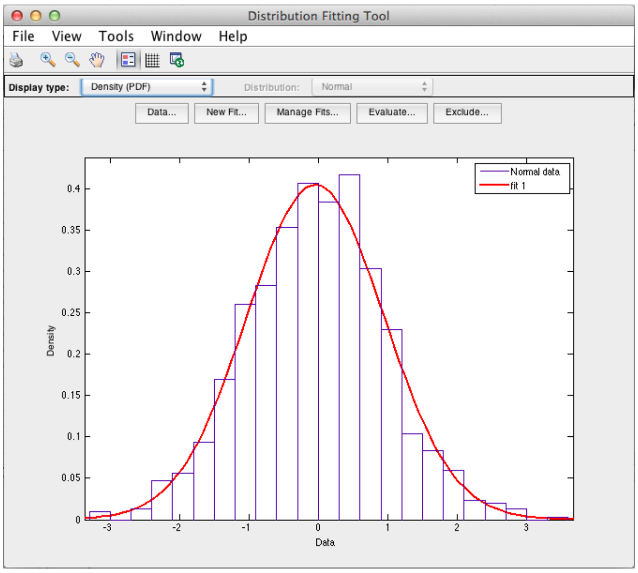
Parameters distribution inference

In the last section we described how the data collection has been performed, now we will present the procedure we followed for the data analysis. Our main goal now is the inference of a suitable PDF for each parameter, which will be used as template function during the OD generation; the synthetic parameters, in fact, will be extracted from the so evaluated distributions.

As we previously mentioned, not all the parameter estimations were eligible to be used for the PDF estimation. The NLLS criterion did not produce a reliable result for some of them, so with a preliminary filtering step the parameters with too high uncertainty (CV> 10%) have been eliminated from the dataset, as well as the outliers value. From now on only the cleaned data will be considered for the PDF inference. The idea now is to represent the histogram of every set of estimations ($\hat{p}$) and to find the distribution which produces the best fit. In order to achieve this goal we took advantage of the MATLAB® Distribution Fitting App. It is a complete toolbox with which a user can easily import data from the MATLAB® Workspace, fit many available custom distributions to the data and display them over the histograms. Among the several pre-built distributions we have, for instance, Normal, Log-Normal, Logistic, Log-Logistic, Gamma and many others. A user-friendly GUI allows also to create exclusion rules for the outliers, to select the rule for the bin dimension and to evaluate and compare different fits. Figure 3.20 shows the main windows of the app, which can be opened with the command `dfittool`. In this example the histogram of 1000 synthetic random numbers sampled from a Normal distribution has been fitted with a Gaussian PDF with $\hat{\mu} = -0.037$ and $\hat{\sigma} = 0.988$, this estimation produces a Log-Likelihood of -1406.39 (we recall that the syntax $\hat{P}$ means *estimated* P). In our procedure, number and dimension of bins have always been defined by the Freedman-Diaconis rule, which says that the optimal bin size can be evaluated as $2IQR(x)n^{-1/3}$, where x is an input data vector containing n samples and $IQR(x)$ is its interquartile range. The Table 3.6 summarises the results we obtained for the parameters in the red and green channels; this table lacks of x_0, y_0, x_0^* and y_0^*, the reason is that the final goal of this work is the development of a tool which lets the

	R	G	B
h_01	9.779	7.049	2.852
h_02	12.958	11.536	8.577
h_03	9.805	5.585	2.346
h_04	17.739	16.224	10.510
h_05	11.008	7.113	3.442
h_06	11.399	11.672	8.708
h_07	17.1674	9.948	3.774
h_08	10.979	11.440	6.789
h_09	9.800	9.578	6.177
h_10	11.122	8.575	3.676
h_11	8.887	9.953	5.41
h_12	10.062	10.387	2.875
h_13	13.068	12.598	5.863
h_14	4.248	7.386	4.703
h_15	8.609	7.659	5.708
dr_01	7.819	7.682	4.135
dr_02	14.128	7.597	2.694
dr_03	5.516	7.905	6.485
dr_04	12.861	10.576	5.296
dr_05	15.658	11.707	6.722
dr_06	11.698	7.437	4.301
dr_07	11.864	9.221	6.468
dr_08	7.937	5.380	1.637
dr_09	12.073	15.337	9.656
dr_10	7.7548	5.348	1.924
dr_11	11.685	7.912	5.914
dr_12	8.373	11.0510	5.176
dr_13	13.927	15.454	11.423
dr_14	11.978	11.449	7.645
dr_15	13.380	8.906	4.516
AVG	11.109	9.656	5.514

Table 3.5: Mean absolute error.



Display type: Density (PDF)
Distribution: Normal

Figure 3.20: MATLAB® Distribution Fitting App main window.

user create a synthetic retinal data set as well as its correspondent GT, so we decided to generate ODs with a manually user-defined centre (see Appendix 5.2 on page 65 for further details). To correctly understand Table 3.6, we must also specify that *Parameter* 1 ($P1$) and *Parameter* 2 ($P2$) acquire different meaning depending on the specified kind of distribution. In fact they respectively refer to lower and upper bound in case of uniform PDF, or mean (μ) and standard deviation (σ) in case of Normal and Log-Normal PDF. As already said, we fitted both the green and the blue channel using the same parametric surface. In order to always obtain a blue channel visually consistent with the green one, instead of estimating a distribution for its parameters, we estimated the correlation between them and the paired green ones. From graphical inspection we assumed a linear correlation between all the paired data sets, so this procedure means nothing else than estimating the two parameters m and q of the linear model $\hat{p}_B = m\hat{p}_G + q$ for each parameter of the blue channel. The obtained results are shown in Table 3.7.

Optic Disc synthesis

We defined the followed procedure to obtain a suitable PDF for each parameter of the two proposed model. In this section we will present the OD synthesis algorithm. The first step is the generation, for the red and green channel, of a set of synthetic parameters extracted from the relative estimated distributions and, for the blue channel, the evaluation of them with the linear correlation model. Being $[\underline{p}_R, \underline{p}_G, \underline{p}_B]$ the generated parameters, the second step is evaluation of the three surfaces $[f_R(x, y, \underline{p}_R), f_{GB}(x, y, \underline{p}_G), f_{GB}(x, y, \underline{p}_B)]$ with the models stated in Equations 3.11 and 3.12. The colour image created with those synthetic channels has to be integrated with a previously generated retinal background. Even in case of simple superimposition, this step needs the definition of a binary map which clearly defines the OD boundaries. On the other hand, providing a reliable GT is one of

Red channel.			
Par. Name	Distrib.	$P1$	$P2$
z_0	Normal	252.941	5.545
a	Log-Normal	-3.919	2.42×10^{-5}
σ	Log-Normal	4.547	0.188
A	Uniform	30.000	40.000
ω	Log-Normal	0.382	0.324
Green channel.			
z_0	Uniform	71.000	93.000
a	Normal	0.0294	0.012
k	Normal	67.355	11.041
σ^*	Normal	103.052	6.929

Table 3.6: Red and Green channels parameters distribution.

Blue channel.		
Par. Name	m	q
z_0	0.720	17.670
a	0.035	0.066
k	0.941	28.680
σ^*	0.573	39.480

Table 3.7: Red and Green channels parameters distribution.

the goals of our project as well. To create the binary mask we may observe that the parameter σ in the red model f_R (see Equation 3.11) controls the OD radius. In fact, the bigger σ is the larger the OD is. The Chebyshev's inequality guarantees that in any probability distribution no more than $\frac{1}{k^2}$ of the values of the distribution can be more than k SD away from the mean (see Equation 3.21, where X is a random variable with finite expected value μ and non-zero variance σ^2).

$$Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{3.21}$$

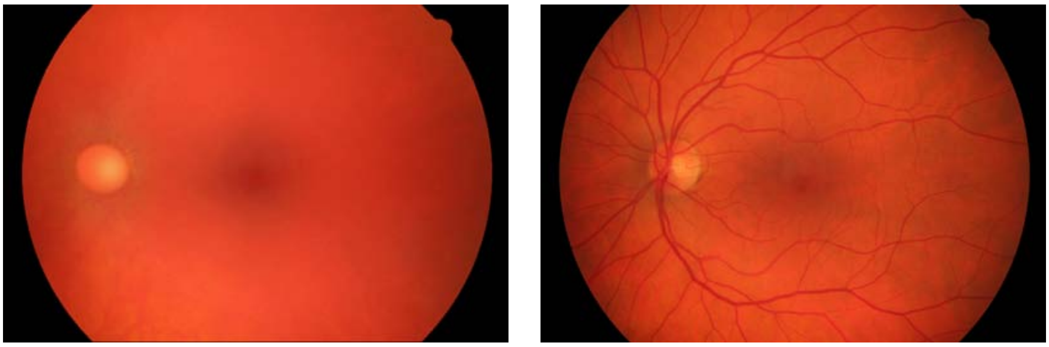
This means that, to create the mask, we basically just decided how many σ away from the mean we wanted to go. This choice has been made looking at the classical Chebyshev's inequality lookup table reported in Table 3.8, where for each k the minimum percentage of the value of the distribution inside the interval $[\mu - k\sigma, \mu + k\sigma]$ and the maximum outside is expressed.

k	min. inside	max. outside
1	0%	100%
1.5	55.56%	44.44%
2	75%	25%
3	88.8889%	11.1111%
10	99%	1%

Table 3.8: Chebyshev's inequality lookup table.

As previously mentioned, all the OD annotations are available to download from the HRF database website. From those data we can see that OD radii (in pixels) are approximately normally distributed around $\mu = 187$ with $\sigma = 14$. We heuristically chose the value of k among the entries of the left column of Table 3.8 selecting the one which generates OD radii similar in dimensions to the real ones. Our choice has been $k = 1.5$ which also guarantees the selection of more than half of the distribution. We are aware that the Chebyshev's inequality, due to the fact that it does not have a priori hypothesis on the distribution, is a poor estimator of the distribution bounds. Actually, because the equation of the distribution is known, an evaluation of the real bounds would have been possible. The reasonable results obtained with the heuristic one made further analysis considered not to be necessary for our purposes.

Once the binary mask has been created, the next step has been to merge of the synthetic ODs with retinal backgrounds. Watching real ODs (e.g. Figure 3.14a and 3.14e) we can see how their edges are not well defined and some blur makes it difficult to accurately individuate the boundaries between OD and background. Merging the two synthetic elements with a simple superimposition using the previously defined binary mask would generate unrealistic sharp edges. In order to replicate the real smoothing effect on the OD edges we used a local Gaussian filter. Figure 3.21 shows an example of the so obtained preliminary result compared to a real RFI (#15 in the HRF healthy subset). At a first glance the retinal phantom looks convincing. The synthetic OD has realistic shape and dimensions and the inner cup, compared to the real one, looks correctly placed with a realistic level of colour intensity. Narrowly



(a) Synthetic retinal background and OD. **(b)** Real RFI.

Figure 3.21: Comparison between synthetic and real RFI.

watching it a non-trivial problem emerges: the jointed results of two different texture techniques, patch-based for the background and model-based for the OD, have completely different level of noise. So, a step of *Quilting-like* noise introduction became necessary. The original Quilting technique has already been described in Section 2.4.2 on page 13. As the reader remembers, this texture synthesis technique generates new images stitching together small patches collected from real images from top left to bottom right corner. Every patch, to be placed, must satisfy some constraint and, to make a visually pleasant transition from one patch to another, its boundaries are cut along the *Best Boundary Cut*. The final image presents two noise components: the first one belongs to the patch themselves and it could be assumed as a classical AWGN; the second one is the noise introduced by the BBC between two neighbouring blocks. The first component is easy to introduce, but in order to make the OD texture consistent with the surrounding background we need to model the second one as well. The Quilting-like noise introduction has been developed following the steps below.

- Random selection, by tossing a coin, of a random number of points inside the OD binary mask.
- Generation of randomly shaped patch centred on each selected point. To generate the patch shape, we started from a 5 × 5 cross-shaped binary matrix.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Then, again by a coin flip, we changed into 1 some of the 0 values, obtaining a new matrix like the following one.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Then a morphological operation that bridges all the 1-valued pixels (which have at least two 0-valued neighbours) has been used. The obtained result could be a matrix like that.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- All those randomly shaped patches have been combined together creating a map like the one depicted in Figure 3.22, where the false colour allows to distinguish among the different patches.

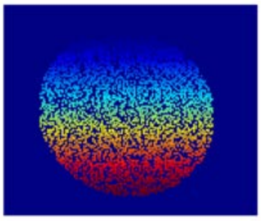


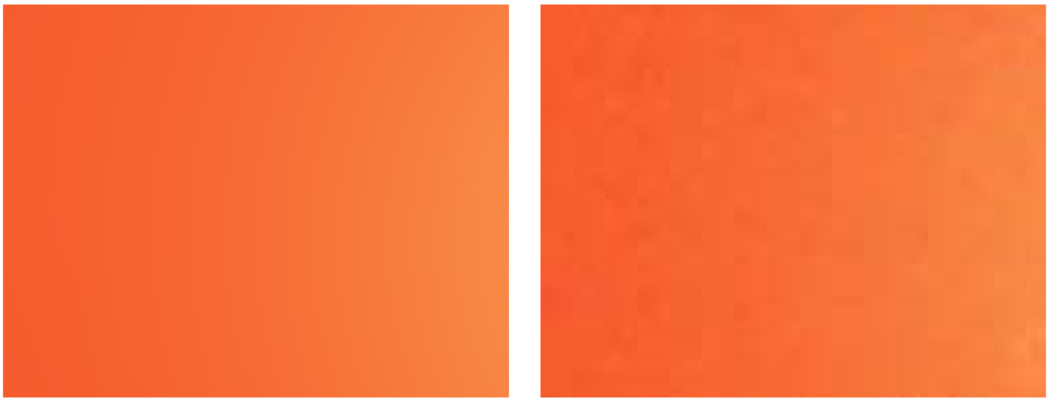
Figure 3.22: Quilting-like noise map.

- As last step a Gaussian noise uniform in each patch has been added.

Usually an AWGN addiction is pixel-based. Intuitively the algorithm described above is a patch-based Gaussian noise addiction. Using Figure 3.23a as input the algorithm produces the noise affected texture shown in Figure 3.23b. We can see how while the figure on the left looks unrealistically smooth, in the one on the right a convincing noisy texture has been synthesised. Some final retinal backgrounds with synthetic OD are depicted in Figure 3.24.

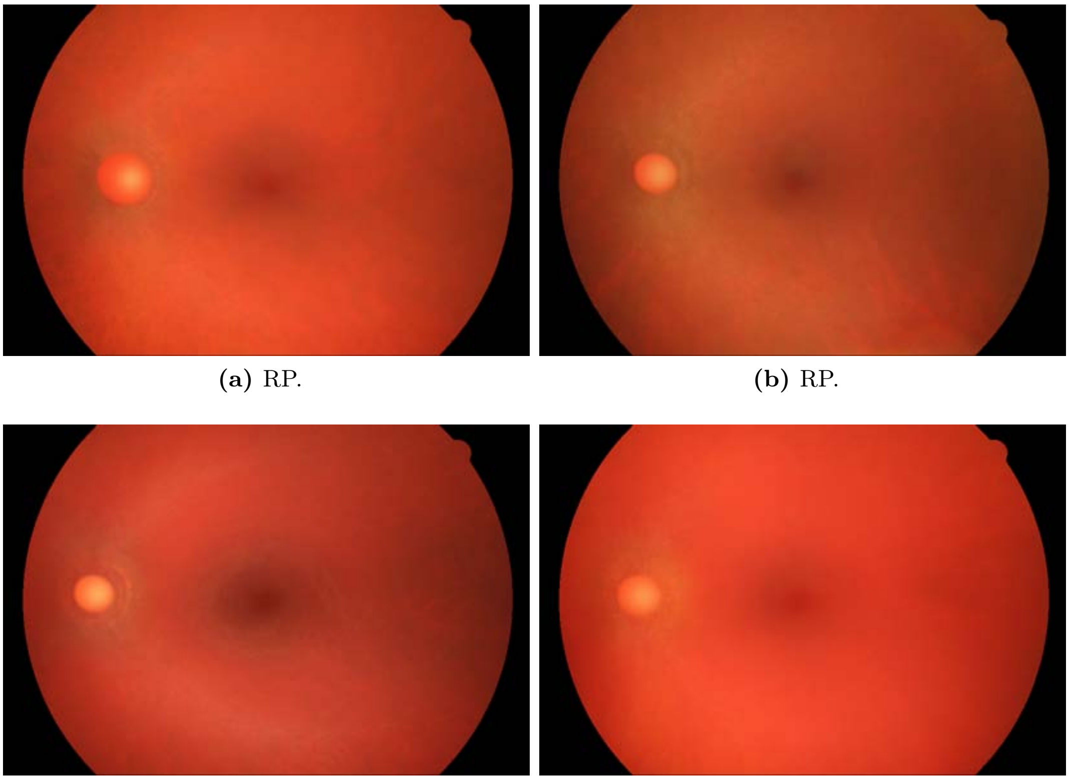
3.4 Chapter Summary

Chapter 3 has been the principal of the thesis. A patch-based texture synthesis technique for the generation of retinal backgrounds has been proposed and described in details. A model-based method for the Optic Disc generation has been introduced as well as the tricks used for the combination of those two elements. Results and comparisons with real images have also been shown for both of the achieved goals and some ideas for future developments have been suggested. In the next Chapter we will describe the quantitative experiments conducted on the results images.



(a) Smooth OD area. **(b)** Quilting-like noise affected OD area.

Figure 3.23: Result of a Quilting-like noise addiction.



(a) RP. **(b)** RP.

(c) RP. **(d)** RP.

Figure 3.24: Synthetic Retinal Backgrounds plus OD.

Chapter 4

Experiments and Results

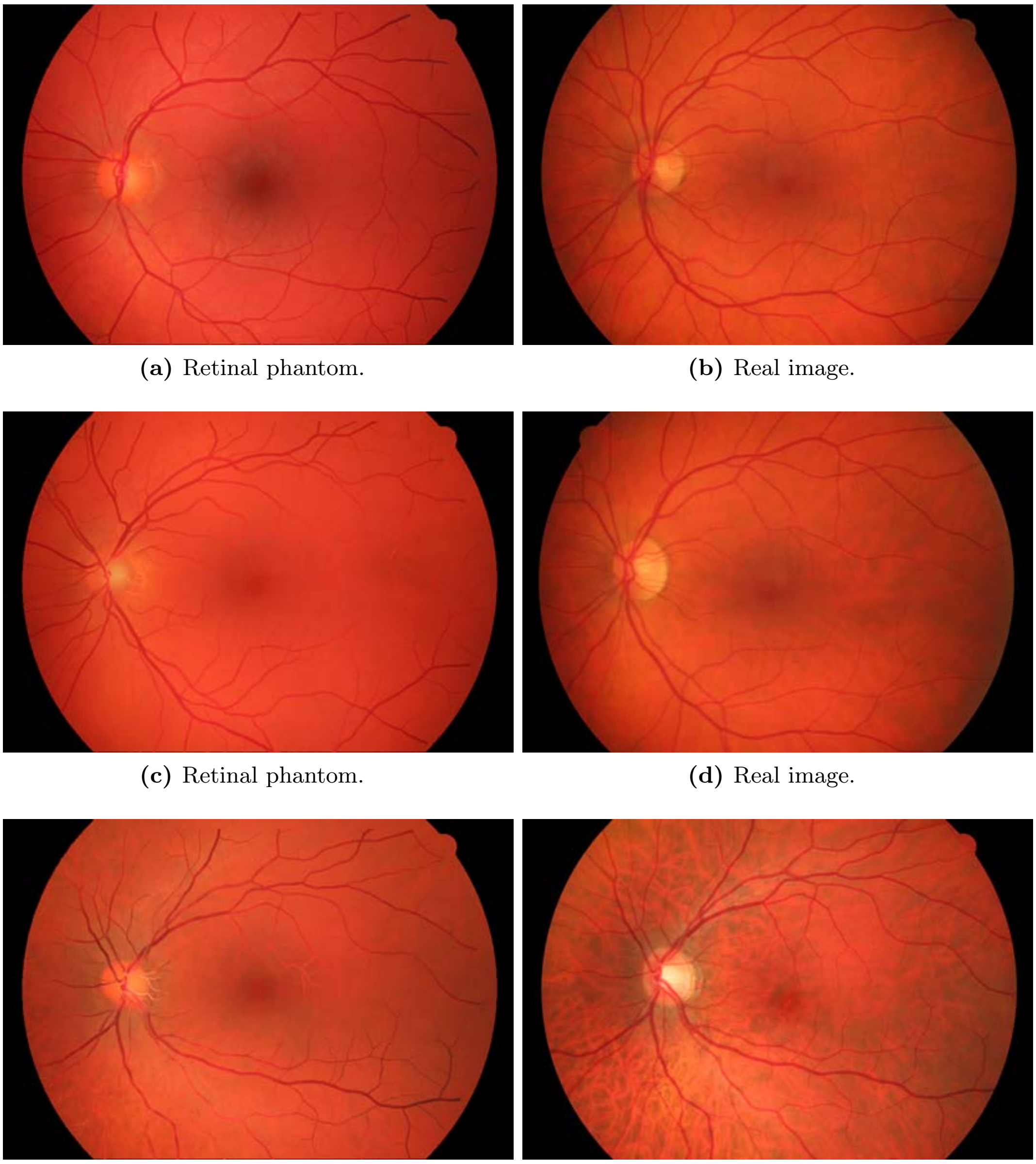
4.1 About this chapter

In this final Chapter the experiments conducted on the synthesised images and relative results will be presented. First of all, we aimed at investigating if our retinal phantoms look like real images in a qualitative and in a quantitative sense. Then, we tried to discover if a typical retinal image analysis algorithm has similar performances when runs on phantoms with synthetic GT and on real images with manual GT. As the reader remembers our main goal is to help the validation step, so we needed a quantitative test to guarantee that our retinal phantoms are not misleading in the evaluation of a typical retinal image analysis algorithms.

4.2 Retinal Phantom Visual Appeareance

The first goal we need to achieve to validate our retinal phantoms is the realistic appearance. This first qualitative observation may seem obvious, but all our work has been conducted trying to simulate the real RFI appearance and satisfying this first requirement is mandatory to give an actual meaning to all the next quantitative tests. Figure 4.1 shows a comparison between three real images (# 06, 05 and 02 in the HRF healthy subset) and three synthetic RFI. As widely outlined this work focused on the generation of non vascular regions. The vessel trees showed in Figure 4.1a, 4.1c and 4.1e are real, in the sense that they have been collected from real images and then, by a procedure similar to the one already described for the synthetic OD (see Section 3.3.2), they have been merged with synthetic backgrounds and ODs. Those real vessel trees have been placed only in order to allow a better visual evaluation of the OD area, such results are obviously not definitive. This work will be complemented in the future with the results from a parallel work-in-progress study on the vessel network synthesis.

From a visual analysis of Figure 4.1 the following observations emerge. The synthetic retinal background looks very realistic; the colour intensity distribution follows the real one, it is brighter in the area surrounding the vessels and it becomes darker towards the extremities. A realistic representation of choroidal vessels is showed in Figure 4.1e as well. The fovea is convincingly represented, as in real images its localisation is not always easy. Looking at the images in the right column, in



(a) Retinal phantom.

(b) Real image.

(c) Retinal phantom.

(d) Real image.

(e) Real image.

(f) Retinal phantom.

Figure 4.1: Comparisons between retinal phantoms and real images.

fact, we can see how fovea and macula are not always well defined and the same consideration emerges from a close observation of the presented retinal phantoms. Our synthetic ODs look promising, they show a reasonable colour intensity, the inner cup is correctly placed in the fovea side and their edge looks realistically merged with the surrounding background. However, to be fair, a better characterisation of the parameters of its models seems to be needed. With the presented method all the synthesised OD looks more similar to the real ones in Figure 4.1b and 4.1d, where the *cupping* effect is less stressed compared with Figure 4.1f. There is still room for further studies to aim at improving this representation of the OD. The real superimposed vessels present evident flaws, they are locally brighter than the background and sometimes they even grow on the fovea. This, of course, is not satisfactory, but a realistic representation of a vessel tree is not on our goals list. We initially decided to add a real vessel network only to allow a better visual inspection of the OD area but unexpectedly the described emerged imprecisions allowed us to sense some of the problems which may be met in the development of synthetic vessel networks.

4.3 Noise level estimation

The first quantitative test we propose is a comparison between an estimation of the noise level in real and in synthetic images. Assuming that the images are affected by additive Gaussian noise, we compared the estimation between its variance σ^2 in real and in synthetic RFI. The evaluation of the two variances has been made taking advantage of the MATLAB® function `evar` developed by D. Garcia and available to download from the author's personal webpage[12] or the MATLAB® CENTRAL File Exchange repository[13]. This function uses a number of thin-plate spline models (presented in [36]) to smooth the image and then assumes that σ^2 can be provided by the model which presents minimum generalised cross-validation score (see [37] for further details).

Running `evar` on the 15 images in the healthy HRF subset we obtained an average $\sigma^2_{real} = 0.0200$, the same function on 15 retinal phantoms returned $\sigma^2_{synth} = 0.0220$. With a two-sample T-test (see the MATLAB® function `ttest2`) on the real and the synthetic variance vectors the null hypothesis H_0, which assumes that the two vectors come from distributions with the same means, has not been rejected at the 5% level (the P-value was 0.3809). The two-sample T-test, however, implies the hypothesis that the tested samples are normally distributed, which is unlikely in our case. So, a Wilkoxon rank sum test has been run as well (with the MATLAB® function `ranksum`). This second test relies on the null hypothesis that the two tested vectors comes from distributions with the same medians. The obtained P-value is 0.2628 and the null hypothesis could not be rejected in this case as well. The Wilkoxon rank sum test implies that the tested samples come from two distribution with the same shape, so a last test to verify if the data collected from both the real and the synthetic images can be assumed as samples drawn from the same underlying continuos population has been evaluated. For this purpose the two-sample Komolgorov-Smirnov test (MATLAB® function `kstest2`) has been selected. This final test did not reject the null hypothesis returning a P-value of 0.1359. The result of the performed statistical tests are summarised by Table 4.1. We can conclude that, assuming that the tested images are globally affected by AWGN, the variance of the noise in the real and in the synthetic images is not significantly different.

Test	P-value	H_0
T	0.3809	acc.
Rank sum	0.2628	acc.
K-S	0.1359	acc.

Table 4.1: Statistical tests results.

With the tests described above we proved that real and synthetic RFI do not have a not significantly different level of noise considering the global images. In order to investigate on the local performances of our Quilting-like tiling technique we evaluated the `evar` function on 100 randomly selected vessel-free windows 50 × 50 pixels wide, half of them belonging to real images and the other half to synthetic ones. After that with a two-sample T-test we tested if those measurements have or not significantly different means. The initial random selection could influence the outcome of the test, so the experiment has beed repeated 10 times. The obtained results are summarised in Table 4.2.

Experim.	P-value	H_0
1	0.332	acc.
2	0.009	rej.
3	0.022	rej.
4	0.084	acc.
5	0.614	acc.
6	0.242	acc.
7	0.756	acc.
8	0.083	acc.
9	0.033	rej.
10	0.418	acc.
AVG	0.259	acc.

Table 4.2: Statistical local tests results.

The results in the table above shows us how in the majority of the circumstances the variance of the noise in the synthetic and in the real retinal backgrounds is not significantly different. This let us conclude that the visual appearance of our synthetic backgrounds is realistic and no significant quantitative differences could be observed.

[12] http://www.biomecardio.com/matlab/evar.html

[13] A link for `evar.m` on MATLAB® CENTRAL.

4.4 Vessel detection

As previously outlined our work aims also at generating a reliable GT for the validation of retinal image analysis algorithms. In this section we present an experiment conducted aiming to test the reliability of the retinal phantoms in algorithms validation. We validated our results proving that in the non-vascular regions no vessel-like patterns are created by our method. For this test a trustworthy vessel segmentation algorithm has been needed and the supervised method described by Soares et al. in [4] has been selected. The authors released an open source MATLAB® tool provided with a user-friendly GUI which allows to specify all the necessary settings as well as to proceed with the segmentation of retinal images. The Soares' method is also available in the VAMPIRE [29, 28]. The selected technique has been evaluated specifying the following settings: the pixel features used for the classification have been the inverted green channel and 5 Gabor filters have been used for scales $= 5, 10, 15, 20, 25$. As the authors suggest in the original paper, these scales have been chosen in order to span all the possible widths of vessel in our 3504×2336 pixels wide images. The classifier has also been trained on the 15 images of the Glaucoma HRF subset. This experiment aims at investigating on the reliability of our retinal phantoms for the vessel extraction algorithm described above. In order to be used in algorithms validation it must be proven that our retinal phantoms are not misleading during a vessel detection, which means that the same algorithm must have approximately the same performances on our retinal phantoms and on real images provided with manual GT. As the reader remembers, this work focused on the generation of non-vascular regions that, waiting to be completed with the synthetic vessel trees, have been temporary populated with real retinal vessel networks. This test consists in a performance comparison of the vessel detector described above on two different databases. The first is the usual healthy HRF subset and the second is a 15 image set of retinal phantoms (RP).Table 4.3 summarises the average value of the performance parameters separately evaluated for the two datasets, where True Positive and True Negative are respectively abbreviated as TP and TN and the performance parameters are evaluated as follows.

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{4.1}$$

$$\text{Accuracy} = Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

$$\text{Specificity} = Sp = \frac{TN}{TN + FP} \tag{4.3}$$

$$\text{Sensitivity} = Sn = \frac{TP}{TP + FN} \tag{4.4}$$

$$\text{Positive Predictive Value} = PPV = \frac{TP}{TP + FP} \tag{4.5}$$

$$\text{Negative Predicted Value} = PPN = \frac{TN}{TN + FN} \tag{4.6}$$

Param.	HRF	RP	abs. diff.
FPR	0.0097	0.0112	0.0005
Ac	0.9540	0.9576	0.0037
Sp	0.9903	0.9888	0.0015
Sn	0.6615	0.6891	0.0276
PPV	0.8932	0.8764	0.0168
NPV	0.9593	0.9648	0.0055

Table 4.3: Performance comparison.

Comparing the HRF and the RP column of the Table 4.3 we can see how our retinal phantoms do not significantly influence the performances of the tested vessel segmentation algorithm. On the last column the absolute difference is reported and, as we can see, it is aways less than 0.03. No intensity adjustment has been provided for the vessel network, so the most meaningful row of Table 4.3 is the first one, in fact t it only concerns the false positive values. The two obtained false positive rate values (FPR) are really close each other and using a two-sample Komolgorov-Smirnov test no significant difference at the 5% level has been observed. Then taking into account the Ac parameter, we can see how the difference is just 0.0037. Running a two-sample T-Test on the Ac values obtain from real and synthetic retinal images no significant difference can be observed (at the usual 5% level). This confirms how our initial goal of generating a synthetic dataset with reliable GT for the validation of retinal image analysis algorithm has been actually achieved.

4.5 The tiling algorithm potentiality

The proposed tiling algorithm has been presented in Section 3.3.1, the last experiment we conducted aims at understanding if our technique could find applications even outside the retinal image analysis field. In fact we tried to synthesise a retinal image using a tiles dictionary populated with actual retinal blocks (they belong to the HRF healthy subset) and as CM a non-retina related image. The obtained result is shown in Figure 4.2.

The usual retinal background in this case has been replaced with a synthetic planet Mars. As we can see in the areas where the actual martian colour intensity was similar to some available tiles the result is reasonable (upper part of the image), in the darker area, instead, strong artefacts are recognisable. This image is nothing else than a funny joke, but it also let us sense a possible application of the presented method to other branches of the Computer Vision field.

4.6 Chapter Summary

In this Chapter we described the conducted tests on the retinal phantoms which aim at understanding the real applicability of our synthetic images in the validation step of retinal image analysis algorithms. The results we obtained confirmed how,

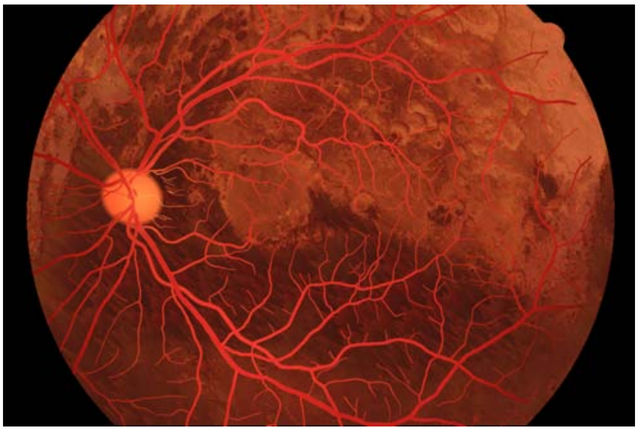


Figure 4.2: Tiling algorithm potentiality test result.

with the technique proposed in this thesis, a trustworthy set of synthetic images could be generated. A detailed user guide of the developed software is available in Appendix A on page 65.

Chapter 5

Conclusions

5.1 Overview and Achieved Results

In this thesis a novel texture synthesis technique which aims at synthesising retinal phantoms has been presented in details. The generated retinal fundus images are similar in characteristics to a given dataset, but their textural anatomical features can be modified in order to simulate a wide range of parameters as well as different populations.

Our method combines together patch-based and model-based technique. The presented patch-based algorithm (page 28) is an improvement of the well-known image Quilting technique by Efros and Freeman [24], while, to our best knowledge, there are no works on literature which report a model-based texture synthesis method similar to the one presented in this thesis (page 30).

This work mainly focused on the generation of non-vascular regions, which means that with the presented technique everyone can easily generate convincing retinal backgrounds, foveae and ODs. The visual similarity between our retinal phantoms and real images has been qualitative proved in Section 4.2. Retinal image analysis algorithms validation is the main application our synthetic images could find and, in order to prove they reliability in this delicate field [38], quantitative tests have been conducted in Section 4.3 and 4.4. From the obtained results we concluded that the proposed retinal phantoms have a not significantly different level of noise with respect to real images, both from a global and a local point of view. The other decisive result we obtained has been the not significantly different level of performance of a typical retinal image analysis algorithm when comparing its evaluation on the proposed retinal phantoms and on real images (see Section 4.4). This confirmed us how the initial goal of providing synthetic retinal images as well as a reliable absolute GT has been achieved.

The whole project has been implemented in MATLAB® and an intuitive GUI has also been developed in order to guide a user into the workflow and allow him to easily generate retinal phantoms. A user guide is available in the Appendix on page 65.

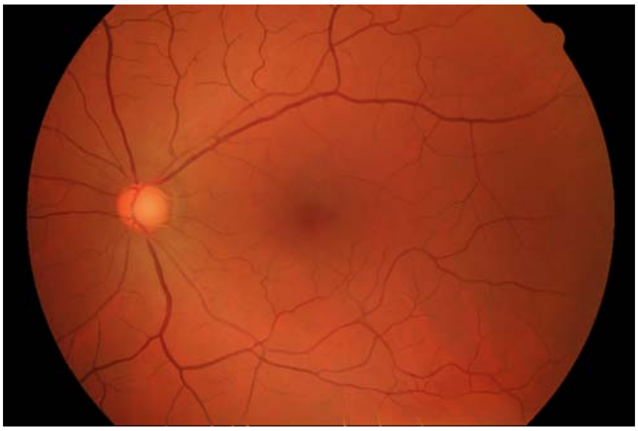


Figure 5.1: Retinal phantom.

5.2 Future Work

With the presented techniques a user can synthesise convincing retinal images in their non-vascular regions. The results shown in this thesis will be completed with synthetic vascular trees generated by suitable fractal models which are currently under development. When all the elements will be combined together a retinal phantoms database will be publicly released. An example of image in this database will look like Figure 5.1.

During the discussion of this thesis, every point that shows potential improvement has been highlighted. In order to be actually deployable, almost every algorithm needs an optimisation step. Fort instance, the algorithm we used for the evaluation of the Manhattan distance can be outperformed by a technique like the one presented in [32]. However, the real bottleneck of all the project is the MATLAB® function used for the generation of the background. During the development of this project we always worked in full resolution (3504×2336 pixels); another simple idea to reduce the elapsed cpu-time could be to work in a downscaled version of the images and then upscale again. Test on the results so obtained should be evaluated in order to verify that the error introduced during the downscaling and upscaling step is negligible. Another important improvement could be the automatic definition of the thresholds used in the background synthesis step. As the reader remembers, in the tiling algorithm, every tile to be placed must satisfy some constraints which always imply a hard thresholding. An automatic detection of those thresholds could not only speed up the code, but even extend the kind of images at which this project is addressed. With the presented work, in fact, we can only recreate retinal fundus photographs; it could be very interesting an extension of this project that aims at generating synthetic SLO images.

Another idea is the extension of this project to images with recognisable diseases. While lesions like exudates and drusens should be developed from scratch, with this technique glaucomatous OD are already possible to synthesise. A population study on the OD models parameters could be interesting in order to understand which value of them are directly correlated with healthy and glaucomatous patients.

The last hint for a possible future work could be the use of the models presented for the synthesis of the OD for its detection. The idea of using a 3-D model for the OD detection has already been proposed by Lowell et al. in [20], but it has been done with a 3-D LoG model with images converted in greyscale. A possible idea could be to separately use the information carried by the three channels to estimate centre, radius and edges of the optic nerve head.

Appendix - The MATLAB GUI

In order to guide the user into the retinal image synthesis workflow, a simple MATLAB® GUI has been implemented. In this Appendix, a detailed description of all the necessary steps for the synthesis of novel retinal phantoms will be provided along with an overview of the developed MATLAB® code. This GUI has been tested and optimised for 64-bit architectures. Working on high resolution images may cause *out of memory* errors when running on 32-bit machines.

5.3 The main window

The main window of the GUI could be launched running the command `RIS` on a MATLAB® Command Window (see Figure 5.2). This is divided into three sections: *a*) the *Preliminary Steps*, *b*) the *Synthesis* itself and *c*) the *Synthetic Images* section where the results are displayed. This separation follows the proposed algorithm workflow described in Chapter 3. In order to synthesise a new image, the user must start with section *a*. Attempting to access at the buttons in section *b* before having satisfied the preliminary steps will generate an error handled with a dialog box. So, in the next section we will describe use and behaviour of the three buttons in section *a*.

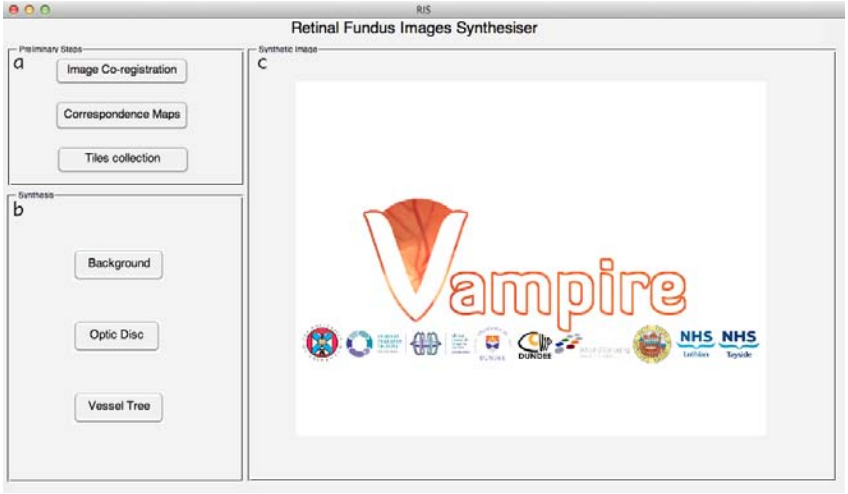
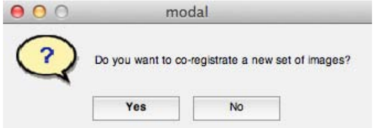


Figure 5.2: Main window on startup.

5.4 Preliminary Steps

5.4.1 The Co-registration process

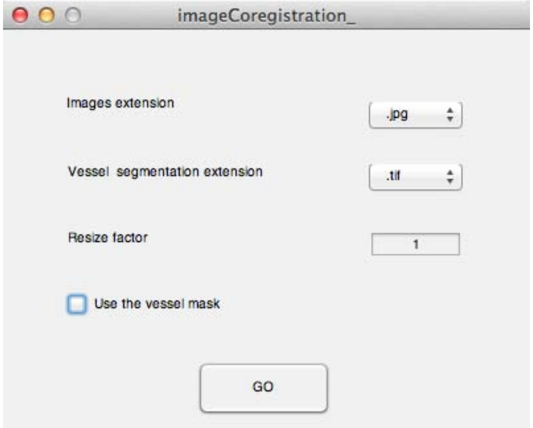
As the box diagram in Figure 3.3 shows, the first step for the synthesis of retinal images is the evaluation of CMs. Their generation needs a set of co-registered real retinal images which can be easily created with the *Image Co-registration* button. Every time an user clicks on a button which corresponds to a cpu-intensive task, a *modal* windows pops up allowing him to decide if run the task again (maybe on a new set of images), or use a previously evaluated result. This is the case of the co-registration process. With the window in Figure 5.3 the user can choose to co-registrate a new set of images, or use a previously co-registrated one.



Do you want to co-registrate a new set of images?

Figure 5.3: The co-registration modal window.

Choosing to co-registrate a new set of images, the user will be prompted with the window in Figure 5.4. A set of retinal images, in order to be co-registrated, needs a correspondent set of binary mask and the *.txt* files generated by the VAMPIRE-Annotation Tool with the coordinate of OD and fovea centre. The binary mask could identify, for instance, the vessel tree or the OD segmentation.



Images extension .jpg

Vessel segmentation extension .tif

Resize factor 1

□ Use the vessel mask

Figure 5.4: The co-registration main window.

With the two popup menus in Figure 5.4 the user can select images and vessel binary masks extensions among the most common ones. If the GUI is running on a 32-bit architecture it may be useful to downscale the images before the co-registration step. For this purpose an appropriate *Resize factor* (< 1) can be specified. The checkbox in Figure 5.4 allows to co-registrate a set of images where the regions identified in the binary map are overshadowed. This could be useful as preliminary step to a successive foreground inpainting. After the first parameters are specified, clicking on the button *GO* the user will be finally asked for the selection of input data and output directory, then the co-registration process starts. At the end of the process the set of co-registrated images will be available in the specified output directory. On the machine used for the development of this project, the co-registration process do not take more than 2 minutes to successfully co-registrate 15 images in HRF full resolution (3504 × 2336).

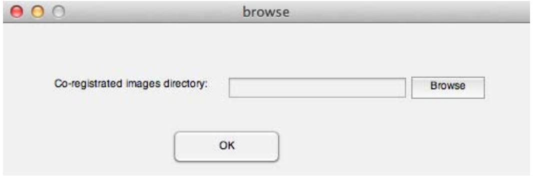


Figure 5.5: The co-registration input folder selection.

In case the user selected to not co-registrate a new set of images, but to generate a CM with already co-registrated ones, the window in Figure 5.5 appears. The user can manually specify the absolute path of the folder containing the co-registrated images, or he can use the *Browse* button to choose it using a standard folder selection GUI.

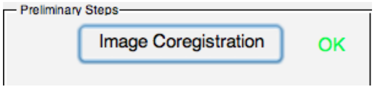


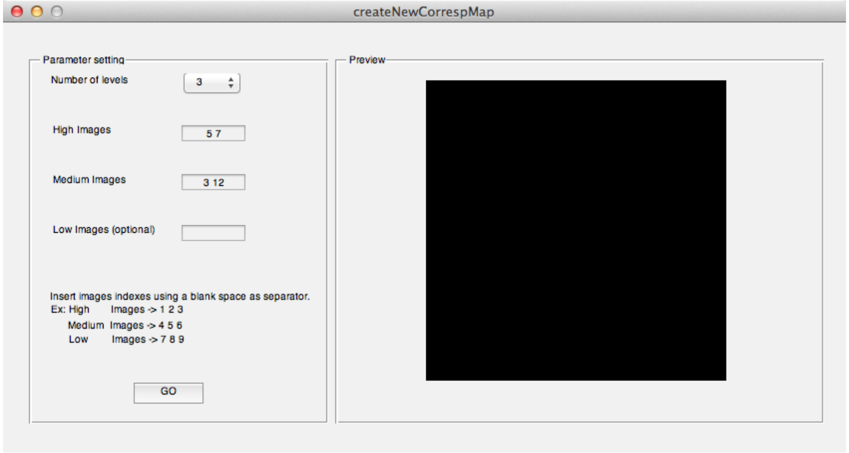
Figure 5.6: The co-registration process is successfully completed.

Once the images have been successfully co-registrated, or a path with already co-registrated ones has been selected, a green *OK* appears next to the *Image Co-registration* button (see Figure 5.6).

5.4.2 The Correspondence Map generation

When a set of co-registrated images is available, the generation of a CM becomes finally possible. As already described for the image co-registration step, the user can choose between generating of a new CM, or selecting a previously evaluated one. Clicking on the *Correspondence Maps* button, a modal window similar to Figure 5.3 appears. Selecting to not generate a new CM the user must specify where to find another one using a window similar to Figure 5.5. When the user chooses to generate a new CM the window in Figure 5.7 pops up. As we can see the window is split into two sections: *Parameter Setting* and *Preview*, both of them will require a user's interaction. The CM generation algorithm has been previously described in Section 3.3.1 on page 21. As the reader remembers, the CMs are generated by weighted averaging of a user defined set of co-registrated images. The elements in the *Parameter Setting* section aim at defining the weight vector. The weight vector statement is important, in fact, different weight vectors correspond to different synthetic retinal backgrounds. The *Number of levels* popup menu allows the user to specify with how many weight levels he want to define the weight vector: 1, 2 or 3.

Selecting 1 implies that all the images have the same weight and the CM become more smooth, selecting 2 the user can specify to which images assign a *high* level and selecting 3 also some *medium* level ones can be specified.



Number of levels: 3

High Images: 5 7

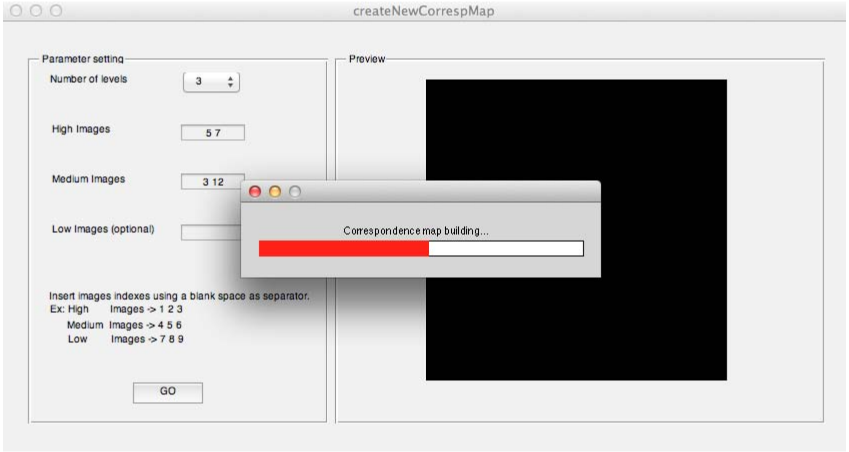
Medium Images: 3 12

Low Images (optional):

Insert images indexes using a blank space as separator.
Ex: High Images -> 1 2 3
Medium Images -> 4 5 6
Low Images -> 7 8 9

Figure 5.7: The Correspondence Map generation main window.

After the selection of the number of levels, the user can specify which weight assign to each image of the selected dataset using the proper input dialog boxes. Usually the name of the images in a specified dataset, contains an increasing number (e.g. the HRF healthy images are called *01_h.jpg, 02_h.jpg, ..., 15_h.jpg*). So, the weight can be assigned specifying those numbers into the input dialog boxes. When the weight vector has been defined, the CM generation could start clicking on *GO*.



Number of levels: 3

High Images: 5 7

Medium Images: 3 12

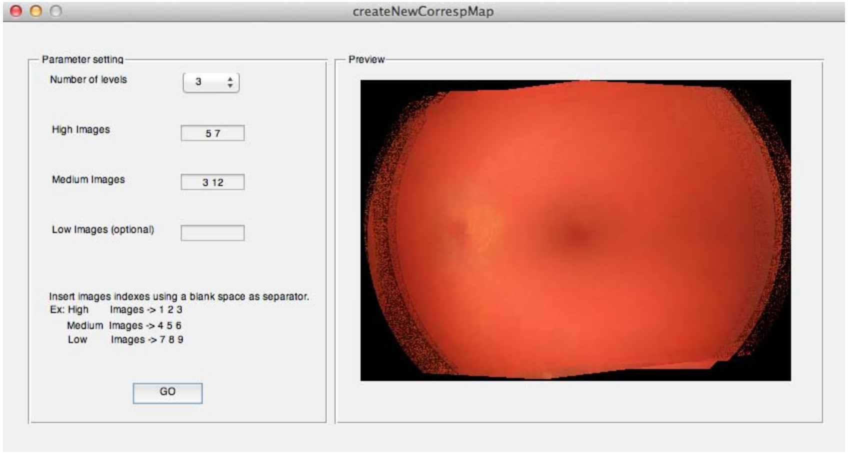
Low Images (optional):

Insert images indexes using a blank space as separator.
Ex: High Images -> 1 2 3
Medium Images -> 4 5 6
Low Images -> 7 8 9

Correspondence map building...

Figure 5.8: Correspondence Map generation in progress.

On the machine used for the development of this project, the generation of a single CM averaging 15 full resolution HRF images takes approximately 40 minutes. When the CM generation process ends, in the *Preview* section a preliminary result appears (see Figure 5.9). We may notice that the generated CM shows some evident flaws toward its periphery.


createNewCorrespMap

Parameter setting

Number of levels: 3

High Images: 5 7

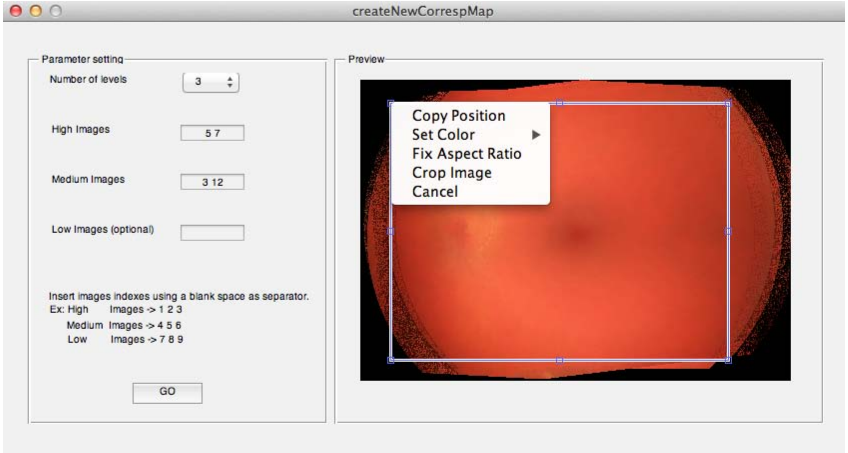
Medium Images: 3 12

Low Images (optional):

Insert images indexes using a blank space as separator.
Ex: High Images -> 1 2 3
Medium Images -> 4 5 6
Low Images -> 7 8 9

Figure 5.9: Correspondence Map preliminary result.

At this point the user is asked to manually crop the CM in order to exclude the peripheral areas (see Figure 5.10).


createNewCorrespMap

Parameter setting

Number of levels: 3

High Images: 5 7

Medium Images: 3 12

Low Images (optional):

Insert images indexes using a blank space as separator.
Ex: High Images -> 1 2 3
Medium Images -> 4 5 6
Low Images -> 7 8 9

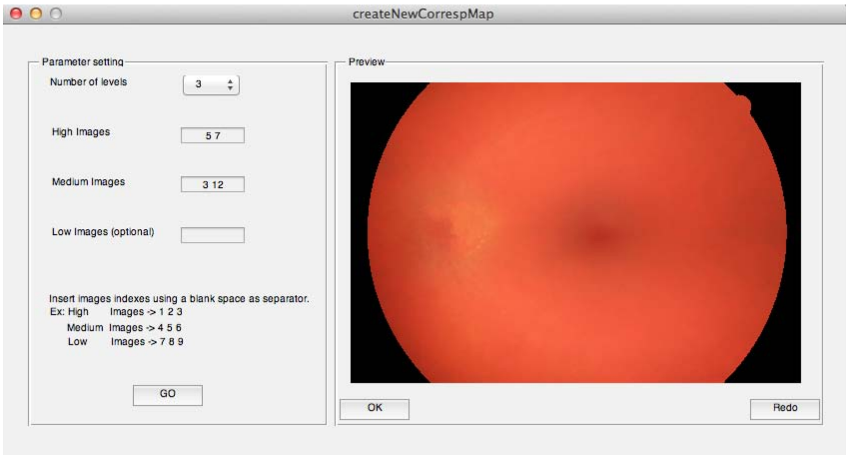
Figure 5.10: Correspondence Map manual cropping.

After the map cropping, the raw results are replaced by the final CM. This operation can be repeated, using the *Redo* button, until the result is satisfactory. The final CM can be saved as *.png* file clicking on the *OK* button (see Figure 5.11).

Once a satisfactory CM has been saved, or an already generated one has been selected, a green *OK* appears next to the *Correspondence Maps* button (see Figure 5.12).

5.4.3 Tiles collection and Dictionary creation

The last step before the actual retinal image synthesis is the collection of a suitable set of vessel-free tiles. This goal can be achieved using the *Tiles collection*



Number of levels: 3

High Images: 5 7

Medium Images: 3 12

Low Images (optional):

Insert images indexes using a blank space as separator.
Ex: High Images -> 1 2 3
Medium Images -> 4 5 6
Low Images -> 7 8 9

Figure 5.11: Correspondence Map result.

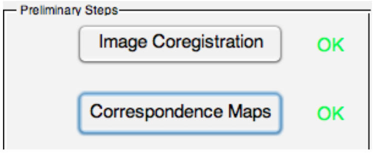
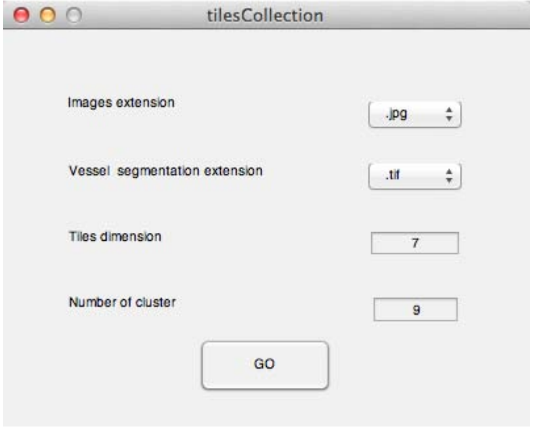


Figure 5.12: A CM has been specified.

button (see the section *a* in Figure 5.2). Along the same line of the previous task, the tiles collection and the relative dictionary creation could not start until the previous steps have been accomplished. Clicking on the *Tiles collection* button a modal window similar to Figure 5.3 pops up. If the user selects to not generate a new tiles dictionary, he will be asked to specify a *.mat* file containing an already generated one. Only files generated by our software (or with the same internal structure) are recognised as suitable tiles dictionaries. Wrong file specification will be handled by an appropriate error dialog box. Depending on the input images, those *.mat* files could be really heavy (from few MBs up to some GBs), so once the file is selected the user might wait for some seconds while the MATLAB® `load` routine runs. After that the files browsing windows automatically closes and the user can continue to use the main GUI. Instead, when the user chooses to collect a new tiles dictionary, the window in Figure 5.13 appears.

The two popup menus on top of the window (Figure 5.13) allow to specify input images and vessel binary maps extensions. The input dialog boxes allow the user to select both the tiles dimension and the K-Means cluster cardinality. As already said in Section 3.3.1 (page 23) the default tiles dimension is 7 × 7 pixels. In regard to the K-Means cardinality, with the Akaike Information Criterion (see Equation 3.6) the identified optimal number was 4 (page 25). However, conducted tests demonstrated how using a higher number of clusters the background synthesis algorithm runs faster. This could be intuitively explained as follows: a light overfit in the clustering process (i.e. a higher number of cluster) implies lower number of tiles in each cluster, lower intra-cluster variability and higher inter-cluster variability. In this scenario the tiling



Images extension .jpg

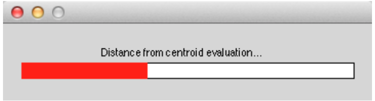
Vessel segmentation extension .tif

Tiles dimension 7

Number of cluster 9

Figure 5.13: The co-registration main window.

algorithm can likely find the best match with less attempts. Because of that the default number of tiles in this dialog box (also used for the current project) is 9. Once the tiles dictionary is created (or selected), the system automatically evaluates a cluster-CM consistent with the previously defined CM.



Distance from centroid evaluation...

Figure 5.14: A CM has been specified.

When all the processes have been completed, the last green *OK* appears next to the *Tiles collection* button (see Figure 5.15) and the system is ready for the synthesis of novel retinal images.

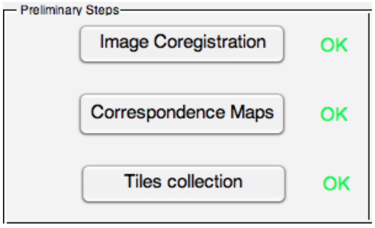
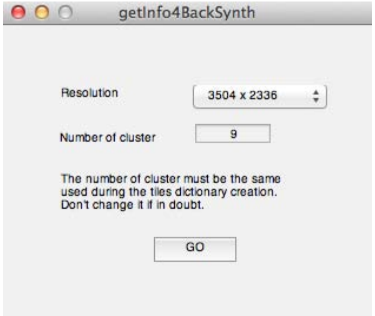


Figure 5.15: A CM has been specified.

5.5 The retinal image synthesis

5.5.1 The background generation

Once the preliminary steps are completed the retinal image synthesis become finally possible. The first element the user needs is a background. The background synthesis algorithm is the real bottleneck of the whole project, on the machine used for the development of this software, up to 15 hours could be necessary to synthesise a 3504 × 2336 pixels background in step of 7 × 7 pixels tiles. Because of that, clicking on the *Background* button the user is asked to choose if generate a new image or specify an already synthesised one. If the user selects to synthesise a new background the window in Figure 5.16 appears. With the popup menu the user can select the resolution of the output image, while the input text dialog box allows to specify the number of cluster in the tiles dictionary. This second option may be redundant (it must be the same specified during the tiles dictionary creation) and it will be deleted in future releases.



getInfo4BackSynth

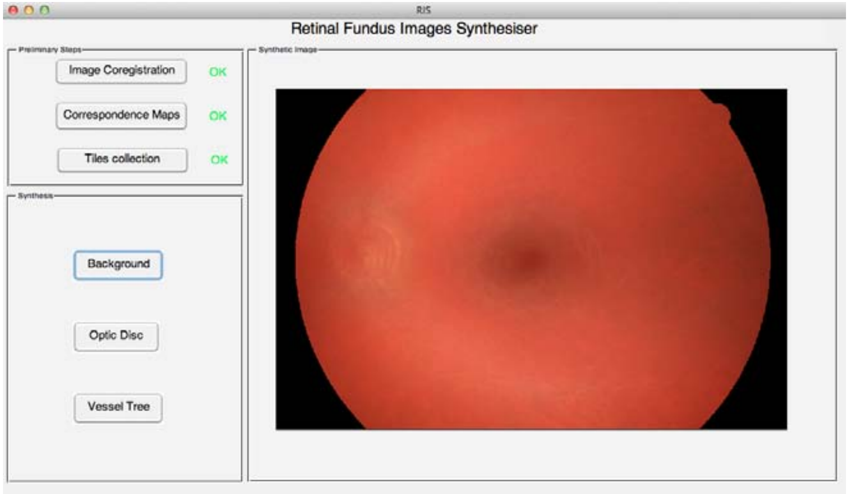
Resolution: 3504 x 2336

Number of cluster: 9

The number of cluster must be the same used during the tiles dictionary creation. Don't change it if in doubt.

Figure 5.16: The background synthesis window.

When the background synthesis is completed, or a previously generated background is selected, the result is shown in the *Synthetic Image* section of the main window (see Figure 5.17). Zoom in/out features are enabled clicking on the image, or scrolling the mouse wheel.



RIS

Retinal Fundus Images Synthesiser

Preliminary Steps

Image Coregistration: OK

Correspondence Maps: OK

Tiles collection: OK

Figure 5.17: Main window after the background synthesis.

5.5.2 The OD generation

Once the background is generated, the last significant feature implemented in this project is the OD synthesis. Clicking on the *Optic Disc* button the user can manually select the OD centre clicking on the desired location in the *Synthetic Image* section. If, accidentally, more than one point is selected, the system will take into account only the first one, ignoring all the successive points. In future releases the manual centre selection may become optional. An example of OD centre manual selection is depicted in Figure 5.18.

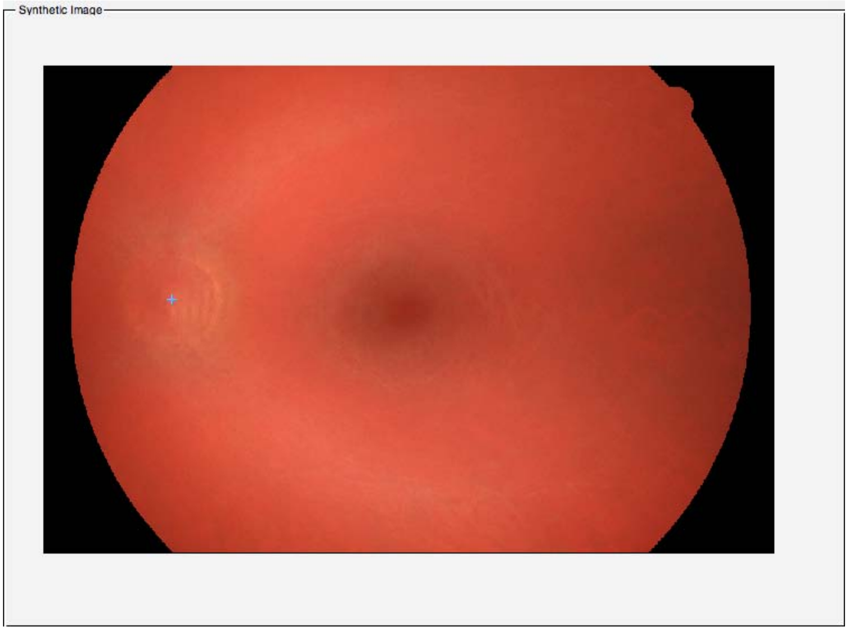


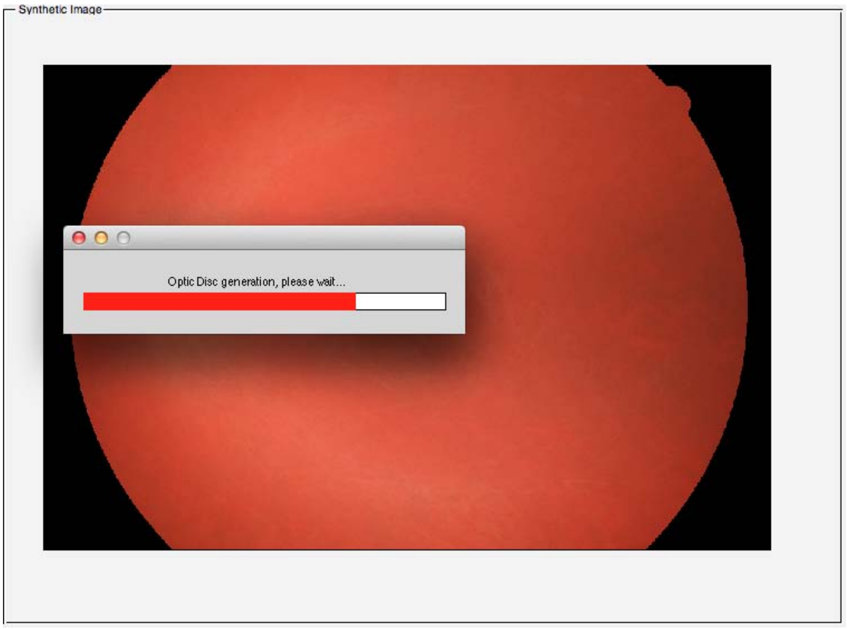
Figure 5.18: OD centre manual selection.

Pressing the *Return* key the OD centre selection is confirmed and the OD synthesis process immediately starts (see Figure 5.19).

An example of generated OD is shown in Figure 5.20; as we can see its colour intensity distribution is acceptable, but, magnificatiing the image, we can sense how the level of noise in the background and in the OD is significantly different. At this level, in fact, the synthetic OD appears too smooth. From Figure 5.20 we can notice that, when the OD creation is completed, two new buttons appear in the bottom left and right corner of the *Synthetic Image* section. If the obtained OD is not satisfactory, a new one can be generated pressing the *Redo* button (bottom-right). Otherwise, if the obtained result is good, the user can confirm the OD pressing *OK* (bottom-left).

As described in Section 3.3.2, in order to make the OD level of noise consistent with the background, a Quilting-like noise is added. This step starts immediately when the user confirms the OD clicking on *OK* (see Figure 5.21).

When the noise component is added, the synthetic OD is finally shown in the *Synthetic Image* (see Figure 5.22) section.



Optic Disc generation, please wait...

Figure 5.19: OD synthesis process.

5.5.3 The vessel network inclusion

The last button in the *Synthesis* section of the main window (Figure 5.2) is called *Vessel Tree*. As we said this work focuses only on the generation of non-vascular regions, but it is waiting to be complemented with a parallel study on the generation of structure and texture of the vessel network. In this release the *Vessel Tree* button only provides a superimposition of a real vessel network, collected from one of the HRF images, on the generated retinal phantom. No intensity adjustment is provided and, in oder to merge the vasculature with the background as realistically as possible, a simple local Gaussian smoothing is provided at the edges. The obtained, temporary, result is shown in Figure 5.23.

5.6 Overview of the MATLAB® code

The retinal images synthesiser MATLAB® package, released with the presented thesis, presents everything the user needs to synthesise convincing retinal backgrounds ad ODs. The use of the developed GUI has been described above. The code has been developed in a modular way and, in case someone wants to contribute in further improvements, a global overview of the package may be helpful. So, taking advantage of the `m2html` tool[14], a dependency graph (shown in Figure 5.24) has been generated.

[14] `m2html` is available to download at http://www.artefact.tk/software/matlab/m2html/.

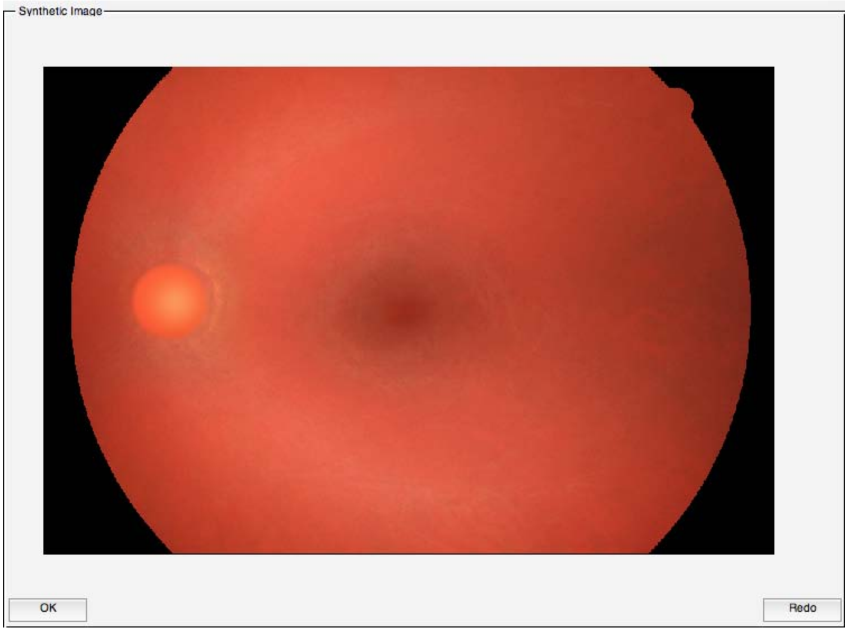
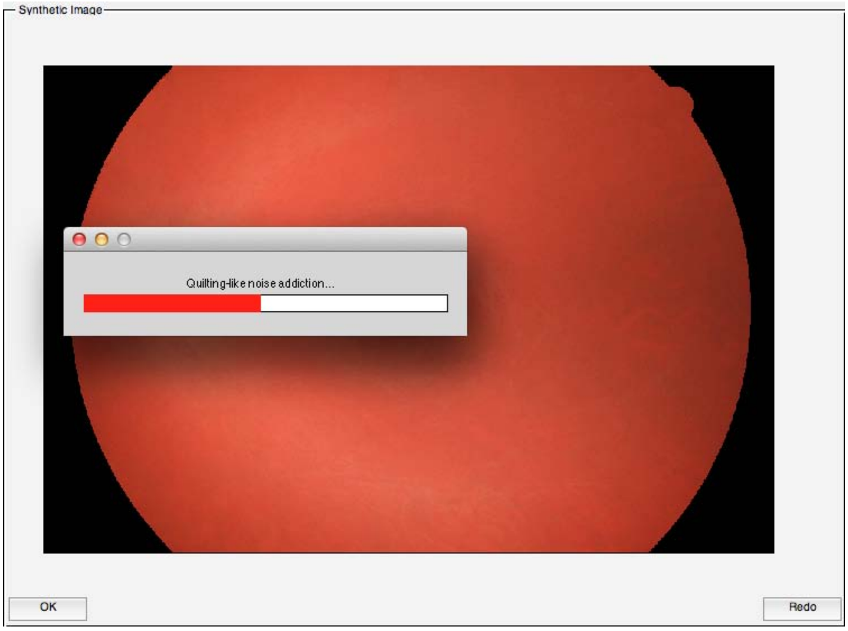


Figure 5.20: Preliminary synthetic OD.



Quilting-like noise addiction...

Figure 5.21: Quilting-like noise addiction in the OD area.

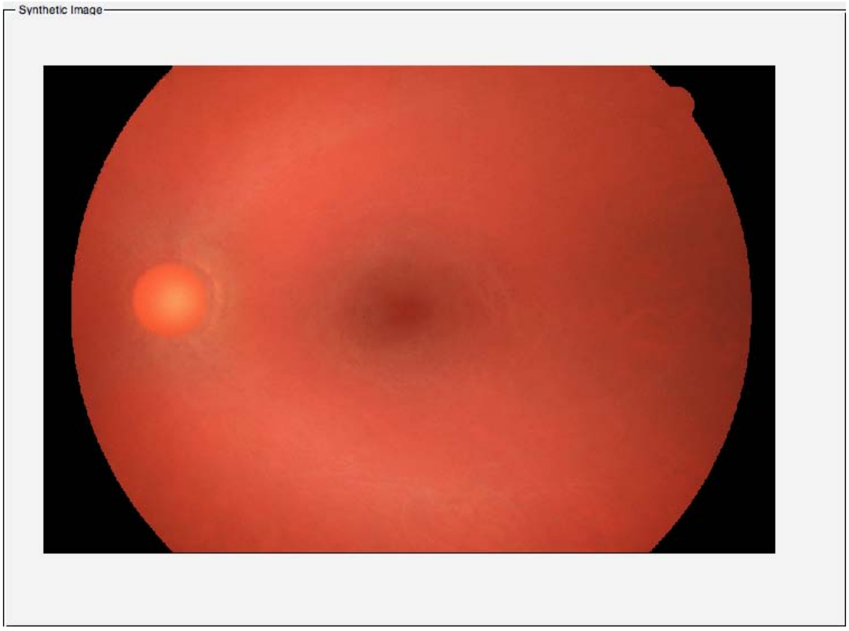


Figure 5.22: The final synthetic OD.

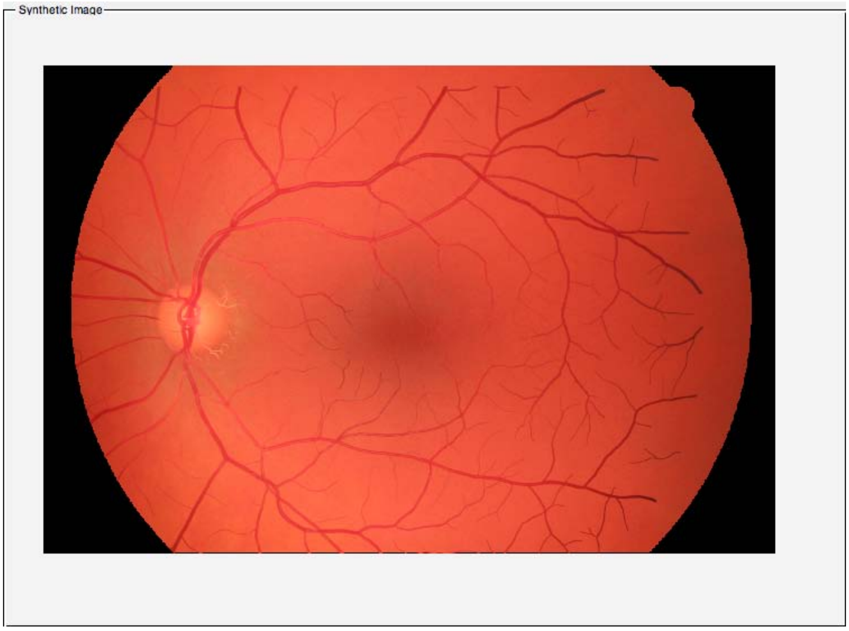


Figure 5.23: A final synthetic retinal fundus image.

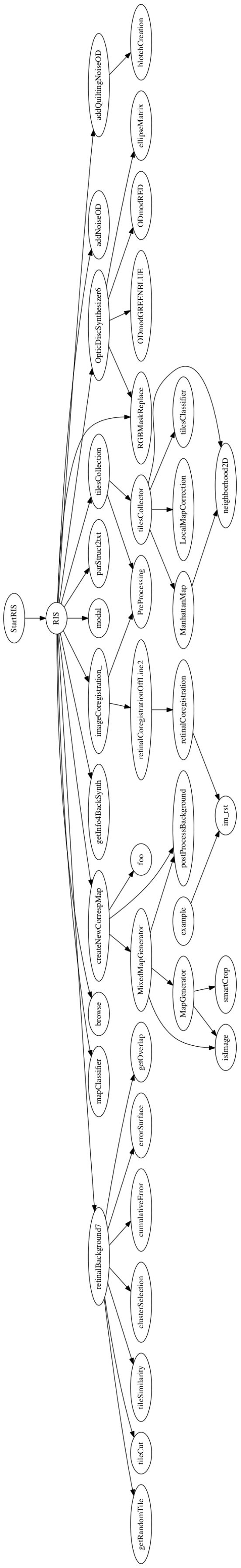


Figure 5.24: The MATLAB® code structure.

Bibliography

[1] H. Jelinek and M. Cree. *Automated Image Detection of Retinal Pathology.* Taylor & Francis, 2010. ISBN: 9780849375569. URL: http://books.google.co.uk/books?id=6DY1AAAACAAJ (cit. on p. 1).

[2] D. Pascolini and S. P. Mariotti. "Global estimates of visual impairment: 2010". In: *British Journal of Ophthalmology* 96.5 (2012), pp. 614–618 (cit. on p. 1).

[3] Y. Wang, G. Ji, P. Lin, and E. Trucco. "Retinal vessel segmentation using multiwavelet kernels and multiscale hierarchical decomposition". In: *Pattern Recognition* (2013) (cit. on pp. 1, 32).

[4] J. Soares, J. Leandro, R. Cesar, H. Jelinek, and M. Cree. "Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification". In: *Medical Imaging, IEEE Transactions on* 25.9 (2006), pp. 1214–1222. ISSN: 0278-0062. DOI: 10.1109/TMI.2006.879967 (cit. on pp. 1, 18, 32, 57).

[5] D. Relan, T. MacGillivray, L. Ballerini, and E. Trucco. "Retinal vessel classification: sorting arteries and veins". In: () (cit. on p. 1).

[6] C. A. Lupaşcu, D. Tegolo, and E. Trucco. "Accurate estimation of retinal vessel width using bagged decision trees and an extended multiresolution Hermite model". In: *Medical image analysis* 17.8 (2013), pp. 1164–1180 (cit. on p. 1).

[7] D.-Y. Yu and S. J. Cringle. "Oxygen Distribution and Consumption within the Retina in Vascularised and Avascular Retinas and in Animal Models of Retinal Disease". In: *Progress in Retinal and Eye Research* 20.2 (2001), pp. 175–208. ISSN: 1350-9462. DOI: http://dx.doi.org/10.1016/S1350-9462(00)00027-6. URL: http://www.sciencedirect.com/science/article/pii/S1350946200000276 (cit. on p. 2).

[8] H. Kolb, E. Fernandez, and R. Nelson. "Webvision". In: (1995) (cit. on p. 2).

[9] M. D. Abràmoff, M. K. Garvin, and M. Sonka. "Retinal imaging and image analysis". In: *Biomedical Engineering, IEEE Reviews in* 3 (2010), pp. 169–208 (cit. on p. 3).

[10] J. Odstrcilik, R. Kolar, A. Budai, J. Hornegger, J. Jan, J. Gazarek, T. Kubena, P. Cernosek, O. Svoboda, and E. Angelopoulou. "Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database". In: *IET Image Processing* 7.4 (2013), pp. 373–383 (cit. on pp. 3, 4, 17).

[11] E. Trucco, A. Ruggeri, T. Karnowski, L. Giancardo, E. Chaum, J. P. Hubschman, B. al-Diri, C. Y. Cheung, D. Wong, M. Abràmoff, et al. "Validating Retinal Fundus Image Analysis Algorithms: Issues and a Proposal". In: *Investigative Ophthalmology & Visual Science* 54.5 (2013), pp. 3546–3559 (cit. on pp. 3, 4).

[12] P. Jannin, J. M. Fitzpatrick, D. Hawkes, X. Pennec, R. Shahidi, M. Vannier, et al. "Validation of medical image processing in image-guided therapy." In: *IEEE Transactions on Medical Imaging* 21.12 (2002), pp. 1445–9 (cit. on p. 3).

[13] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. "Design and construction of a realistic digital brain phantom". In: *Medical Imaging, IEEE Transactions on* 17.3 (1998), pp. 463–468 (cit. on p. 7).

[14] L. A. Shepp and B. F. Logan. "The Fourier reconstruction of a head section". In: *Nuclear Science, IEEE Transactions on* 21.3 (1974), pp. 21–43 (cit. on pp. 7, 8).

[15] H. M. Gach, C. Tanase, and F. Boada. "2D - 3D Shepp-Logan phantom standards for MRI". In: *Systems Engineering, 2008. ICSENG'08. 19th International Conference on.* IEEE. 2008, pp. 521–526 (cit. on p. 7).

[16] E. Hoffman, P. Cutler, W. Digby, and J. Mazziotta. "3-D phantom to simulate cerebral blood flow and metabolic images for PET". In: *Nuclear Science, IEEE Transactions on* 37.2 (1990), pp. 616–620 (cit. on p. 7).

[17] B. Aubert-Broche, M. Griffin, G. B. Pike, A. C. Evans, and D. L. Collins. "Twenty new digital brain phantoms for creation of validation image data bases". In: *Medical Imaging, IEEE Transactions on* 25.11 (2006), pp. 1410–1416 (cit. on p. 7).

[18] M. A. Sagar, D. Bullivant, G. D. Mallinson, and P. J. Hunter. "A virtual environment and model of the eye for surgical simulation". In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* ACM. 1994, pp. 205–212 (cit. on pp. 8, 9).

[19] G. François, P. Gautron, G. Breton, and K. Bouatouch. "Image-based modeling of the human eye". In: *Visualization and Computer Graphics, IEEE Transactions on* 15.5 (2009), pp. 815–827 (cit. on pp. 8, 10, 11).

[20] J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, E. Fletcher, and L. Kennedy. "Optic nerve head segmentation". In: *Medical Imaging, IEEE Transactions on* 23.2 (2004), pp. 256–264 (cit. on pp. 9, 10, 32, 63).

[21] A. Criminisi, P. Pérez, and K. Toyama. "Region filling and object removal by exemplar-based image inpainting". In: *Image Processing, IEEE Transactions on* 13.9 (2004), pp. 1200–1212 (cit. on pp. 11–13, 21).

[22] S. Tae-o-sot and A. Nishihara. "Exemplar-based image inpainting with patch shifting scheme". In: *Digital Signal Processing (DSP), 2011 17th International Conference on.* IEEE. 2011, pp. 1–5 (cit. on p. 11).

[23] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. “Image inpainting”. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 2000, pp. 417–424 (cit. on p. 11).

[24] A. A. Efros and W. T. Freeman. “Image quilting for texture synthesis and transfer”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 341–346 (cit. on pp. 11, 13, 14, 20, 28, 61).

[25] W. G. Backhaus, R. Kliegl, and J. S. Werner. *Color vision: Perspectives from different disciplines*. Walter de Gruyter, 1998 (cit. on p. 14).

[26] L. D. Griffin and A. Sepehri. “Performance of CIE94 for nonreference conditions”. In: *Color Research & Application* 27.2 (2002), pp. 108–115 (cit. on p. 14).

[27] P. Guobing, X. Fang, and C. Jiaoliao. “A Novel Algorithm for Color Similarity Measurement and the Application for Bleeding Detection in WCE”. In: *International Journal of Image, Graphics and Signal Processing (IJIGSP)* 3.5 (2011), p. 1 (cit. on pp. 15, 29).

[28] A. Perez-Rovira, T. MacGillivray, E. Trucco, K. Chin, K. Zutis, C. Lupascu, D. Tegolo, A. Giachetti, P. Wilson, A. Doney, et al. “VAMPIRE: vessel assessment and measurement platform for images of the REtina”. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE. 2011, pp. 3391–3394 (cit. on pp. 18, 57).

[29] E. Trucco, L. Ballerini, D. Relan, A. Giachetti, T. MacGillivray, K. Zutis, C. Lupascu, D. Tegolo, E. Pellegrini, G. Robertson, et al. “Novel VAMPIRE algorithms for quantitative analysis of the retinal vasculature”. In: *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*. IEEE. 2013, pp. 1–4 (cit. on pp. 18, 57).

[30] A. Giachetti, K. S. Chin, E. Trucco, C. Cobb, and P. J. Wilson. “Multiresolution localization and segmentation of the optical disc in fundus images using inpainted background and vessel information”. In: *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE. 2011, pp. 2145–2148 (cit. on pp. 18, 32).

[31] A. Meijster, J. B. Roerdink, and W. H. Hesselink. “A general algorithm for computing distance transforms in linear time”. In: *Mathematical Morphology and its applications to image and signal processing*. Springer, 2000, pp. 331–340 (cit. on p. 23).

[32] A. Hoover, V. Kouznetsova, and M. Goldbaum. “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response”. In: *Medical Imaging, IEEE Transactions on* 19.3 (2000), pp. 203–210 (cit. on pp. 24, 62).

[33] M. Muhr and M. Granitzer. “Automatic Cluster Number Selection Using a Split and Merge K-Means Approach”. In: *Database and Expert Systems Application, 2009. DEXA ’09. 20th International Workshop on*. Aug. 2009, pp. 363–367. DOI: 10.1109/DEXA.2009.39 (cit. on p. 25).

[34] D. Pelleg, A. W. Moore, et al. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters.” In: *ICML*. 2000, pp. 727–734 (cit. on p. 25).

[35] G. H. Ball and D. J. Hall. “A clustering technique for summarizing multivariate data”. In: *Behavioral science* 12.2 (1967), pp. 153–155 (cit. on p. 25).

[36] M. Buckley. “Fast computation of a discretized thin-plate smoothing spline for image data”. In: *Biometrika* 81.2 (1994), pp. 247–258 (cit. on p. 55).

[37] D. Garcia. “Robust smoothing of gridded data in one and higher dimensions with missing values”. In: *Computational Statistics & Data Analysis* 54.4 (2010), pp. 1167–1178 (cit. on p. 55).

[38] E. Trucco and A. Ruggeri. “Towards a multi-site international public dataset for the validation of retinal image analysis software”. In: *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE. 2013, pp. 7152–7155 (cit. on p. 61).

List of abbreviations

Ac Accuracy

AIC Akaike Information Criterion

AWGN Additive White Gaussian Noise

BBC Best Boundary Cut

CDR Cup-to-Disc Ratio

CIE International Commission on Illumination

CM Correspondence Map

CT Computed Tomography

CV Coefficient of Variation

DR Diabetic Retinopathy

FOV Field-of-View

FPR False Positive Rate

GT Ground Truth

GUI Graphical User Interface

HRF High-Resolution Fundus

LoG Laplacian of Gaussian

MAE Mean Absolute Error

MR Magnetic Resonance

NLLS Nonlinear Least Squares

OD Optic Disc

PDF Probability Density Function

PET Positron Emission Tomography

RFI Retinal Fundus Image

RP Retinal Phantom

RSS Residual Sum of Squares

SD Standard Deviation

SLO Scanning Laser Ophthalmoscopy

SNR Signal-to-Noise Ratio

SSD Sum of Squared Differences

TN True Negative

TP True Positive

VAMPIRE Vessel Assesment and Measurement Platform for Images of the REtina

This thesis has been written with LaTeX on my MacBook Pro
finished on 26th March, 2014.