

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Tesi di Laurea Magistrale in Bioingegneria

De Biasi Maddalena

Automatic Generation of Retinal Fundus Image Phantoms: Vascular Regions

Relatore: **Prof. A. Ruggeri**
Correlatore: **Prof. E. Trucco**



Luglio 2014

Alla mia famiglia

*Talent just defines what you do. It doesn't define what you are. Deep down,
I mean. When you know what you are, you can do anything.*
Terry Pratchett

Abstract

Retinal fundus imaging has been shown to provide easily accessible biomarkers for high-incidence systemic conditions like diabetic retinopathy, stroke and cardiovascular disease. Biomarker discovery requires computer based image analysis with validate algorithms, and, in turns, validation requires ground truth in the form of significant volumes of images annotated by experts. Obtaining such annotations sets is expensive, laborious, and not always feasible. This study aims to generate realistic retinal fundus colour images (phantoms), similar in appearance to a given dataset, and associated with full ground-truth values. In the synthesised retinal phantoms, textural and anatomical features can be controlled to simulate a wide range of situations, hence different populations and their characteristics. This work is mainly focused on the vasculature region and on its relationship with background, which is generated by a parallel work. The geometry of the vasculature is reproduced with a model based algorithm and the texture is added on the vessel tree with a patch based algorithm. 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® 2013a.

Keywords: Synthetic Vasculature, Retinal Fundus Imaging, Medical Image Phantom, Ground Truth.

Contents

1	Introduction	1
1.1	Relevance and Aim of the work	1
1.2	Anatomy Of The Eye	2
1.3	Validation In Retinal Image Analysis	4
2	Related Work	7
2.1	Previous Work on Synthetic Vasculature	7
2.2	Major Temporal Arcade Model	11
2.3	Bifurcation Model	13
3	Proposed Method	15
3.1	Materials and Software	15
3.1.1	Image database	15
3.1.2	VAMPIRE software suite	16
3.2	Description Of The Method	17
3.3	Vessel Geometry	18
3.3.1	First branching level	19
3.3.2	Second branching level	21
3.3.3	Third branching Level	22
3.3.4	Tortuosity adding	23
3.3.5	Fourth branching Level	25
3.3.6	Curve point connection	28
3.3.7	Radius adding	29
3.3.8	Freeman distance	31
3.3.9	First vessel calibre model	32
3.4	Vessel Texture	37
3.4.1	Mask and maps	37
3.4.2	Illumination correction	41
3.4.3	Tiles collection	43
3.4.4	Linear vessel generation	45
3.4.5	Warping and background hooking	47

4 Experiments and Results	49
4.1 Visual Appearance	49
4.2 Vessel Detection	56
5 Conclusion and Future Work	59

List of Figures

1.1	Anatomy of the eye	2
1.2	Healthy human retina in a fundus image	3
2.1	Result presented by Kocinski et al. [16]	8
2.2	Result presented by Kōkai et al. [14]	10
2.3	Result presented by Sagar et al. [33]	11
2.4	Example of fitted parabola on MTA	12
2.5	Bifurcation diagram	14
3.1	Example image of HRF database	16
3.2	Vessel tree geometry synthesis block diagram	18
3.3	Examples of synthetic images with different levels of tortuosity	25
3.4	Results from curve point connection	28
3.5	Connection and skeletonization of pixels on a curve	30
3.6	Structuring element	31
3.7	Result from dilation	31
3.8	Example of Freeman chain code directions	32
3.9	Example of Annotation Tool usage	33
3.10	Gaussian distribution fit and histograms of first vessel calibre datasets	35
3.11	Examples of vasculature masks created with the proposed algorithm	36
3.12	Tiles collection block diagram	38
3.13	Example of AV classification	39
3.14	Example of mask post processing	39
3.15	HSV colour space	42
3.16	Illumination correction on a image of HRF database with the proposed algorithm	44
3.17	Synthetic retinal image algorithm block diagram	46
4.1	Comparisons between real and synthetic artery trees	50
4.2	Comparisons between real and synthetic vein trees	51
4.3	Comparisons between real and synthetic vasculature geometry	53
4.4	Comparisons between real and synthetic images	55

4.5	Example of VAMPIRE working on a synthetic image	57
5.1	Example of Retinal Fundus Image Phantom	59

List of Tables

3.1	Parameters for the gaussian PDFs, first level	20
3.2	Parameters for the gaussian PDFs, second level	23
3.3	Parameters for the gaussian PDFs, tortuosity	24
3.4	Sample mean and standard deviation of first vessel calibre data	34
3.5	Parameters for the gaussian PDFs, first vessel calibre	37
4.1	Performance comparison	58

Chapter 1

Introduction

1.1 Relevance and Aim of the work

In the last twenty years image processing has been greatly developed in the medical field and this has led to an improvement in the potentialities of such techniques in the ophthalmological research. It has been shown that retinal image analysis plays a significant role in the diagnosis of several pathologies [38][39]. The prevalent diseases that can be studied via eye imaging analysis include glaucoma, diabetic retinopathy, macular degeneration and hypertensive retinopathy from cardiovascular disease. Glaucoma and diabetic retinopathy are the most common causes of blindness in the developed countries; an early diagnosis can in most cases prevent visual loss and blindness in patients with retinal complications [38]. A better understanding of the morphology of the retina is one of the ways to obtain an early diagnosis of cardiovascular and cerebrovascular diseases. Novel image analysis techniques are now available to help clinicians in diagnosis and follow-up procedures. Such techniques are based e.g. on vessel segmentation [35],[41], vessel classification [31] or vessel width estimation [21]. The development of automatic diagnostic systems as well as the availability of relatively low-cost cameras to take direct images of the retina allow these techniques to be used in large-scale screening programs, with significant resource saving.

Automatic systems, however, work through algorithms. Each and every one of such algorithms need to be validated and the validation requires ground truth, i.e. the result we expect the algorithm to return. The purpose of the present work is to create the ground truth through the construction of synthetic images of the back of the eye, in particular of the retina. The modelling has been divided into two parallel projects, aiming to separately describe the background and the vascular system of the interested region. The current work will only focus on the construction of a model for the vessels, but is strictly related to its parallel work and the results are obtained combining the two algorithms. The presented technique has been developed

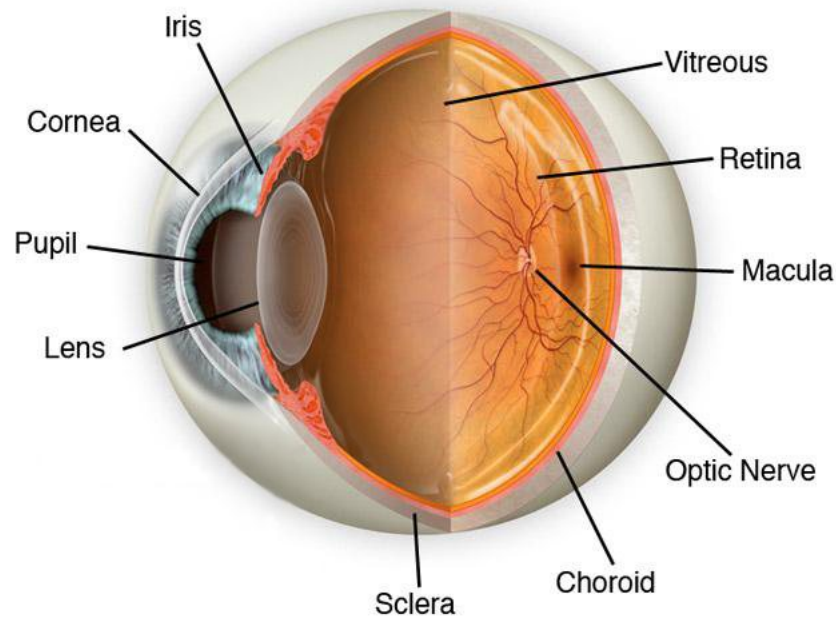


Figure 1.1: Anatomy of the eye ¹

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[®] 2013a. In order to allow the reader to understand the following dissertation, we start with a short explanation of the anatomy of the eye and an introduction to the validation process.

1.2 Anatomy Of The Eye

The eye is an organ that allows to receive light stimuli from the outside world, that will be converted into visual information by the visual apparatus. It has a roughly spheric shape and is protected by three fibrous tunics: moving from the outside to the inside of the eye we meet the sclera, the choroid and finally the retina. The anterior segment of the eye is the part visible from the outside; the sensible part of the eye, where the visual process starts, is instead the posterior segment. The anterior segment is mainly made up by a series of lens structures in sequence, namely cornea, aqueous humor, iris, pupil and crystalline lens. The sclera is the white part of the eye, which becomes transparent in the front gradually fading into the cornea, the first lens from which the light enters the eye. A colored circular muscle, the iris, is placed behind the cornea. Its main function is to control the quantity of

¹<http://www.institutdeloeil.com/eye-treatment/eye-anatomy.html>

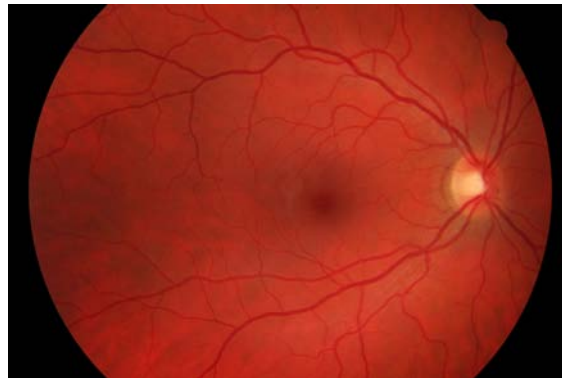


Figure 1.2: Healthy human retina in a fundus image ²

light that enters the posterior segment. In the center of the iris there is the pupil, a black-looking aperture that allows light to enter. Just behind the pupil there is a second lens, the crystalline lens, which is mainly responsible of the light convergence. Between the cornea and the iris and between the iris and the crystalline lens there are two fluid chambers, containing a liquid fluid called aqueous humor. A third fluid chamber, which contains a more dense fluid named vitreous humor, is located behind the crystalline lens, and together with the retina forms the posterior segment of the eye.

The retina contains two different types of photoreceptors, cones and rods, that convert the light into electrical stimuli. The retina is a thin transparent layer divided into two main areas: the central area, named macula, that contains the fovea, and a peripheral area, which is disposed to crepuscular and night vision. Most of cones are concentrated in the macula and especially in the fovea, that is the area most sensible to contrast. Through the two main lenses (cornea and crystalline lens) the light converges into this area. The Optic Disc (OD) is the area on the retina from which the optic nerve departs; it usually has an elliptical shape.

The retina has a complex and articulated vessel network that transports the oxygen. The main artery of the retina is the central artery, that emerges from the OD and immediately divides into an upper and lower branch. This two branches bifurcate into two main arcades, the Major Temporal Arcade (MTA) and the Nasal Arcade. We therefore have four main arcades in the retinal vasculature: the upper MTA and NA and the lower MTA and NA. The vein vasculature follows a similar shape of the artery vasculature. There are no anastomosis between vessels of the same type, in other words arteries do not cross with arteries and veins do not cross with veins. Another particular aspect of the retinal vasculature is that it covers almost the entire retina

²healthy HRF database, see section 3.1.1

except for the macula, which is oxigenated by choroidal vessels. Thus no vessel branching from the central artery and the central vein grow into the macula region.

1.3 Validation In Retinal Image Analysis

As introduced before, the aim of this thesis is to produce ground truth for helping research developing automatic tools for retinal image analysis. The problem of validation in image analysis can be defined as *the process of showing that an algorithm performs correctly by comparing its output with a reference standard* [38]. As it is outlined in [39] the validation process of a retinal image analysis algorithm usually needs four main steps.

- A data sample has to be selected, that has to be representative for the specific validation purposes.
- A reference standard, or ground truth, has to be collected on the selected sample.
- It is necessary to run the algorithms on the sample data.
- The performances of the algorithms have to be compared with the reference standard.

The key step of validation is to quantitatively estimate the agreement of automatic and manual measurements, in other words an algorithm cannot declare a better accuracy than the reference standard. Thus, the reference standard plays a very important role in validation. The main issue of creating a reference standard for validating RIA systems is the variability of the expert judgment; it is a notorious fact that the expert judgment is affected by variability both within the annotation of the same dataset performed by different experts (interobserver variability) and in the annotation of the same dataset made by the same expert in different times (intraobserver variability). Multiple annotations on the same dataset could help having a more trustworthy ground truth, but it is not totally clear how to reconcile multiple reference values, e.g. averages or inter-rater reliability metrics. An aspect of multiple annotations that cannot be neglected, is that very often annotating some features on retinal images can be time requiring, for example the vessel segmentation. The ground truth is not only taken into consideration in a validation area of interest, it is also used in pattern recognition and machine learning, and an imperfect reference standard can be a motivation behind weak learning methods, that are anyways currently rarely used in RIA systems.

For the purpose of generating the ground truth, an alternative to using expert annotation of the sample images could be the creation of eye phantoms

for establishing a true gold standard. Such synthetic images could provide ground truth for every feature of the human retina. For example, for the already mentioned problem of manual vessel segmentation a powerful tool in RIA validation could be having a dataset in which the vasculature is synthesized, so that the location, edges, widths and vessel type classification (artery or vein) is provided with 100 % reliability in every point of the image.

Chapter 2

Related Work

This chapter is divided into two main sections. Aim of the first section is to give an overview on the best known works related to our purpose, describing how they obtain synthetic images and phantoms, especially for vasculature. During the development of this work we had to face some issues, that required the use of models. In this chapter, sections 2.2 and 2.3 are spent to explain the theory, known from literature, behind those models. Details on how models are then implemented and used can be found in the next chapter.

2.1 Previous Work on Synthetic Vasculature

Literature includes different methods for the creation of synthetic vasculature. As a first category, we want to present algorithms which synthesize 3D vasculature. Those methods are usually implemented to investigate the properties of the vasculature in organs and give a quantitative description of the blood flow in the vessel trees ([16], [17], [34], [22]). Other models present in literature use fractals to give a representation of the vessel tree ([20], [14], [40], [36], [33]).

3D vasculature models

A significative example of vascular 3D modeling is presented in [16]. Kocinski et al. give a quantitative description of the blood vessel trees and demonstrate that a relationship exists between the physical parameters of the trees and the numerical texture parameters, computed with the use of vasculature tree images. In order to find such relationship, they use their synthetic images to extract texture parameters for the analysis, with a specialized program named MaZda.

In the considered work a three-dimensional geometric vessel tree is constructed and then converted to images. For this purpose, the following steps are recursively performed: the ROI is divided in voxels, so that there is a

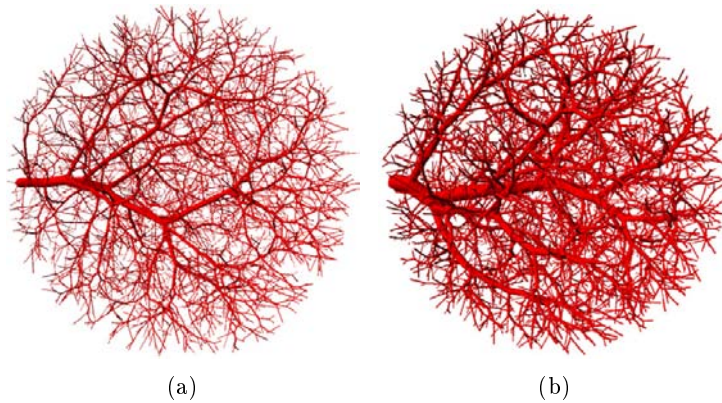


Figure 2.1: Example images of [16]: 2 classes of simulated vasculature trees with different blood viscosity. (a) 1.0 cP (water), (b) 10.0 cP.

uniform probability that each point in the ROI is an end point. A random point inside the ROI is chosen and the distance between the new point and the existing vessels is computed. The new point is connected with the midpoint of the nearest branch. A new segment is added and the previously existing tree is adapted in accordance to the model of blood flow based on the mass preservation principle (see section 2.3). The simulation ends when a preset number of vessels is added. The next step is to convert the tree model into an image. To simulate the grey level the authors rely on the point spread function model of an MRI scanner. The voxels of the image are filled with a contrast-generating medium (the blood) and the signals are proportional to the voxels' volume. Examples of images generated with this method are shown in figure 2.1.

At a superficial visual inspection the quality of the image rendering is satisfying, but a more thorough analysis shows that the vasculature is not completely realistic. The shape of the branches is piecewise linear, the texture added on the vessels is homogeneous and only depends in the quantity of blood contained in each region, and no relationship with the background of the images is taken into consideration. From the analysis of the method, it is also possible to observe that the direction of vessel growth is not controlled, because the probability density function is supposed to be uniform in the entire ROI. For all of this reasons the method cannot be applied for our purposes.

A similar work to the one just presented is [17], in which Kretowski et al. consider the shape of an organ as well as introducing a control of the vascular growth through a birth-death probability in the ROI. At each recursive cycle the entire vasculature is optimized, to respect a bifurcation model. Analogous considerations as those made for Kocinski can be made

for this work.

No further investigations on similar works will be presented. For further details we recommend the previously cited literature.

Fractal vasculature models

A second class of models that can be found in literature for the vasculature synthetic generation is fractal based models. Usually, those models make use of L-Systems. L-Systems were proposed in 1968 by Lindenmayer as a mathematical model for describing cellular interaction during their development. An L-System is a parallel rewriting system consisting in an initiator or axiom and a set of rewriting rules or productions. Starting with the axiom, symbols are converted into new symbols or strings of symbols according to the production, repeating the substitution several times. The strings obtained can be converted (usually with the use of turtle geometry) into spatial objects. The L-System theory has seen several extensions e.g. stochastic, context-sensitive or parametric.

Liu et al. [20] combine stochastic L-System with a version of bifurcation model that, other than radii and angles, also takes into account the length of the vessels to create several types of vessel structures e.g. renal arteries or heart arteries. The rendering of the vasculature is in general quite realistic, but still remains clearly synthetic. The vessel trees are piecewise linear and no texture is added on the vasculature. This is probably due to the fact that the aim of the authors was to generate vasculature for surgery simulators for laparoscopic virtual reality and not reproducing synthetic images for validation.

A very significant work on the modeling of blood vessels using L-System is due to Kókay et al.[14], who model the vessels of the retina. The idea behind [14] is to develop a tool, called GREDEA, for patient specific monitoring programs for examining the blood circulation of the human retina. They use parametric L-Systems with an evolutionary strategy. Since parametric L-Systems evolve, two kinds of evolutionary algorithms are applied: genetic algorithms are used in rewriting rules of the evolved L-System, and evolution strategies are applied in the parameters of the rules.

With the use of a fitness function based on the distance of two images the authors set the L-System parameters. They try to describe the vasculature of a given retinal image. An example of an image used in their work and a result obtained with their method can be found in figure 2.2. This work seems very promising, but, as it is shown in figure 2.2, the level of vasculature complexity that can be reproduced using this method is very simple. To reproduce a more complex vasculature the authors claim that more evolution steps are necessary and the computational time would grow quadratically.

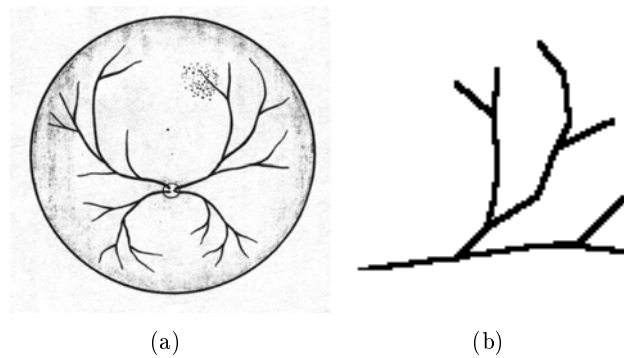


Figure 2.2: Example of retinal image and an L-System that describes it [14].

However the complexity of the retinal image they try to describe is very low and they do not give details on the preprocessing on the retinal image, simply referring to the literature. This method returns results that have a far too low quality for our purposes, but it still gave us some useful hints, e.g. the description of the branches with a set of parameters, related to the shape, the angle and the length of the branch; the division of the retina into four quadrants and sub-trees; and the choice to describe only one of this four.

The last example we want to present is slightly different from those reported above. It is an anatomical detailed 3D computer graphic model of the eye and surrounding face, done by a research team of The University of Auckland, New Zealand, in collaboration with the McGill University of Montreal, Canada, in 1994 [33]. It is designed for surgical simulation, involving both visual and mechanical simulations of features of the human eye coupling computer graphic realism with finite element analysis. They develop this environment to provide simulation for training and to assist during surgery, as a part of a teleoperated micro-surgical robot for eye surgery. The authors are very accurate in describing every detail of the anatomy of the eye, both the exterior and the interior part. The most important part related to our work is their way of describing the retina.

The authors propose a deformed spherical segment as a model for the retina. The Optic Disk is simulated as a yellowish circle from which the blood vessel trees depart. The retinal blood vessels have been generated using a fractal tree. The tree is then projected onto the retinal sphere and a repulsion factor is added in order to keep the large vessels from growing near the fovea centralis (see figure 2.3). The image looks reasonable at a first glance, but after a scrupulous analysis noticeable shortcomings are easy to recognise, for example the aspect of the background is not realistic. The fovea is represented by a smoothed white area lighter than the rest of the background. The vessel tree are piecewise linear and filled with uniform

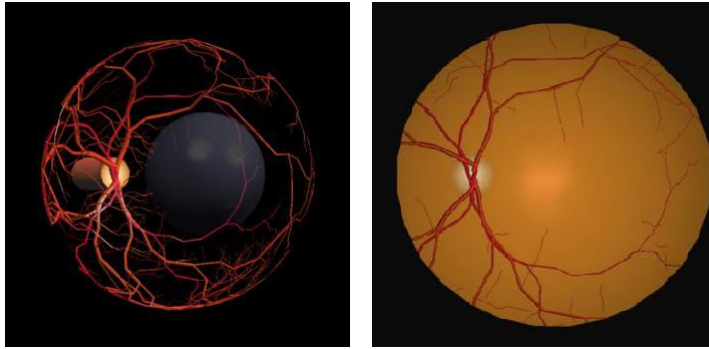


Figure 2.3: Examples of images in [33]

color. Furthermore the number of branches should be higher and some controls like the crossings between vessels of the same type should be added. Taking into account that rendering a convincing retinal synthetic image was not a goal and that that paper was published in 1994, those results are a really good starting point.

2.2 Major Temporal Arcade Model

As we have shortly introduced before, during the development of this work it was necessary to use some models to solve issues. The first model that has been used is the Major Temporal Arcade Model. The problem we had to solve was to find a main shape to be followed by the vessel branches in the retinal images. Many studies using different kinds of models have been taken into consideration ([8], [15], [26], [25]). To extract the MTA from fundus images, for example, Kochner et al. [15] extract edge points on the main blood vessels and then fit an ellipse on these points using the generalized Hough Transform (GHT). Fleming et al. [8] proposed a method to extract the MTA by means of vessel enhancement and semielliptical curve fitting using the GHT. Both of these methods show a good tracking of the MTA, but for this work we have chosen the parabolic model proposed by Oloumi et al. [25] [26] because it is a good compromise between simplicity and ability of describing the shape of the arcades. In their work, the authors define a directional model using the parabolic formulation and assume that the preferred direction of the vessels is tangential to the parabolas themselves. Their proposed method was tested with retinal fundus images from the DRIVE database. The image size in this database is 584x565 pixels with a field of view (FOV) of 45° and spatial resolution of approximately 20 μm per pixel. Their algorithm consists of three main steps: a preprocessing step, a MTA recognition using the ITA and the STA algorithms, and a parabola detection using the GHT method. After the tracing of the vessel tree in

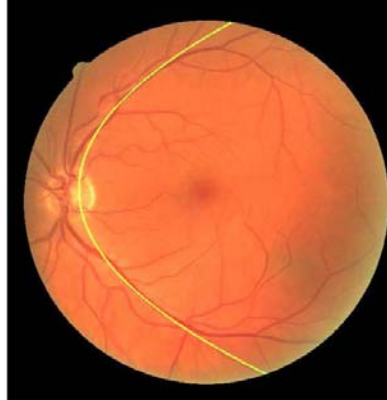


Figure 2.4: Example of fitted parabola on MTA

the FOV, at each branching point the larger branch is followed to define the MTA. They then use this traced MTA combined with the hand-drawn MTA to obtain the data to be fit. A useful hint for our work is taken from Oloumi's work: the MTA is made by the branches with the largest calibre.

The general formula defining a parabola with its directrix parallel to the y -axis and its symmetrical axis parallel to the x -axis is

$$(y - y_0)^2 = 4a(x - x_0) \quad (2.0)$$

where (x_0, y_0) is the vertex of the parabola and the quantity $4a$ is known as latus rectum. They then use the Hough transform, varying a , to detect the best parabolic fitting. The range used for a is restricted by physiological limits on the MTA and the size of the image. The range they use for a , for DRIVE database, is $[35,120]$. In the present work, even if the image phantoms we want to realize have the same size of HRF database, we choose values compatible with such range.

Details on the implementation of this model can be found in section 3.3. As we mentioned before there are many models of Temporal Arcade in literature. We used the simplest to be implemented and the most used in literature, but we need to highlight that the real MTA is diverging from the OD toward the macula, then it converges down into the macular region. If one wants to use a parametric curve to fit a model to entire MTA, it would be difficult to define a specific model; an ellipse appears to be the best estimate to the overall vascular structure but not a specific arcade. Future work could be the use of a integrated parabolic-elliptical model as a compromise between the described models, to exploit the benefits of both.

2.3 Bifurcation Model

The second model necessary for this work is the bifurcation model. The morphology of vascular network have been studied for many decades but, though significant results have been produced, the complex set of physiological properties make it difficult to understand. The first example of modeling the branching network was made by Murray in 1926 [23], but a previous study by Wilhelm Roux in 1878 describes the relationship among the angles and radii of bifurcating blood vessels [32]. Both of the studies show how the relationship between radii and angles arise from some simple principles of optimality. Many studies have been lead since the first works of Rouxe and Murray, but the basic principles remained the same ([37], [28],[11],[2]). In the retina, a large number of bifurcation exist and the principle of optimality can also be used to analyze them as it is shown in [3], [42]. The blood is modeled as a Newtonian fluid. The idea of Murray's model is reduced to the assumption that a physiological vascular network matches the least possible biological work needed for maintaining the blood flow through it at a required level. Applying Navier-Stokes equations and considering the flow of a fluid in a uniform circular cylindrical pipe of radius r and length l , the biological work Ψ can be modeled as:

$$\Psi = \frac{8\eta l J^2}{\pi r^4} + m\pi r^2 l \quad (2.0)$$

In equation 2.3 the two terms represent the cost of overcoming viscous drag during blood motion through the vessels and it is equal to the inverse of Poiseuille's law, and the metabolic energy required to mantain the volume of blood and the vessel tissue.

- J represents the blood flow rate;
- m is the metabolic coefficient
- η is the blood viscosity

We have now to minimize the function, with respect to the radius r :

$$\frac{\partial \Psi}{\partial r} = 0 \quad (2.0)$$

From equation 2.3 we find:

$$J = kr^3 \quad \text{where} \quad k = \sqrt{\frac{m\pi^2}{16\eta}} \quad (2.0)$$

k is constant for the tissue taken in cosideration. As shown in figure 2.5, we can consider valid the conservation principle at the node: $J_0 = J_1 + J_2$.

The branches radii at node comply with the Murray's law:

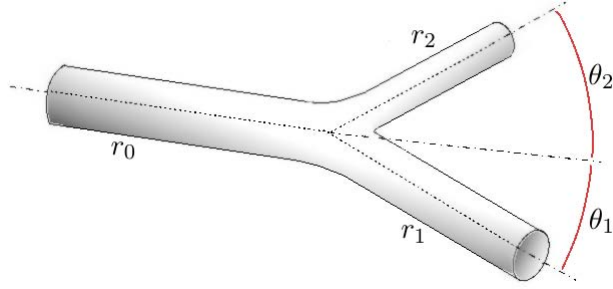


Figure 2.5: Bifurcation diagram

$$r_0^x = r_1^x + r_2^x \quad (2.0)$$

Many values for x are proposed in literature, for example in [11] the value of x is set to $\frac{7}{3}$, in [2] the value is 3. In this work x is set to 3, in accordance to [2], because it seems to better represent the real relationship between angles and radii in the bifurcation. With reference to figure 2.5 for annotations, Murray's law can be rewritten as:

$$r_1 = \sqrt{r_0^2 \cos \theta_1 - r_2^2 \cos(\theta_1 + \theta_2)} \quad (2.0)$$

$$r_2 = \sqrt{r_0^2 \cos \theta_2 - r_1^2 \cos(\theta_1 + \theta_2)} \quad (2.0)$$

From equations 2.3 and 2.3 we can write:

$$\cos \theta_2 = \left(\frac{r_0^2}{r_1^2} - \cos \theta_1 \right) \frac{r_1^2}{r_2^2}$$

Then:

$$\theta_2 = \arccos \left[\left(\frac{r_0^2}{r_1^2} - \cos \theta_1 \right) \frac{r_1^2}{r_2^2} \right] \quad (2.0)$$

From equations 2.3, 2.3 and 2.3 we can write, eliminating the superfluous radius:

$$\cos \theta_1 = \frac{r_0^4 + r_1^4 - (r_0^3 - r_1^3)^{\frac{4}{3}}}{2r_0^2 r_1^2} \quad (2.0)$$

Being the same for the other branch:

$$\cos \theta_2 = \frac{r_0^4 + r_2^4 - (r_0^3 - r_2^3)^{\frac{4}{3}}}{2r_0^2 r_2^2} \quad (2.0)$$

All of these equations will be necessary for the bifurcation model implementation that can be found in section 3.3.

Chapter 3

Proposed Method

3.1 Materials and Software

3.1.1 Image database

As we have introduced before, we are able to synthesize retinal images similar in characteristics to a specific dataset. The dataset we have chosen is the HRF (High-Resolution Fundus image database)¹[5],[24], that is composed of images taken with high resolution. The database was released in 2011 and it is made of 45 images, divided in three different datasets of 15 images each. All of the images were taken in Eye Ophthalmology Clinic Zlin, Czech Republic. In the first dataset are images taken from patients that have no pathological conditions, the second dataset is related to patients with diabetic retinopathy, the third is composed of images taken from glaucomatous patients. In the dataset are also available masks determining the FOV (Field Of View), and the gold standard for segmentation. The gold standard data are generated by a group of experts working in the field of retinal image analysis and clinicians from the cooperated ophthalmology clinics. All the images are acquired with the mydriatic fundus camera CANON CF-60 UVi equipped with CANON EOS-20D digital camera with a 60-degree FOV. The image size is 3504 x 2336 pixels. All the images are encoded with 24-bit per pixel and stored in standard JPEG format with low compression rates.

The pathological conditions of the patients in the last two dataset are clearly visible on the images, especially in the diabetic retinopathy dataset, which contains hemorrhages and leakage of blood and serum. In the diabetic retinopathy new abnormal fragile vessels develop on the surface of the retina and may grow toward the center of the eye, so images taken from pathological patients are not suited to our purposes. We want to reproduce images of healthy retinas, leaving the disease development for future works. For this reason we only take into consideration the first dataset of healthy patients.

¹free to download: <http://www5.cs.fau.de/research/data/fundus-images/>

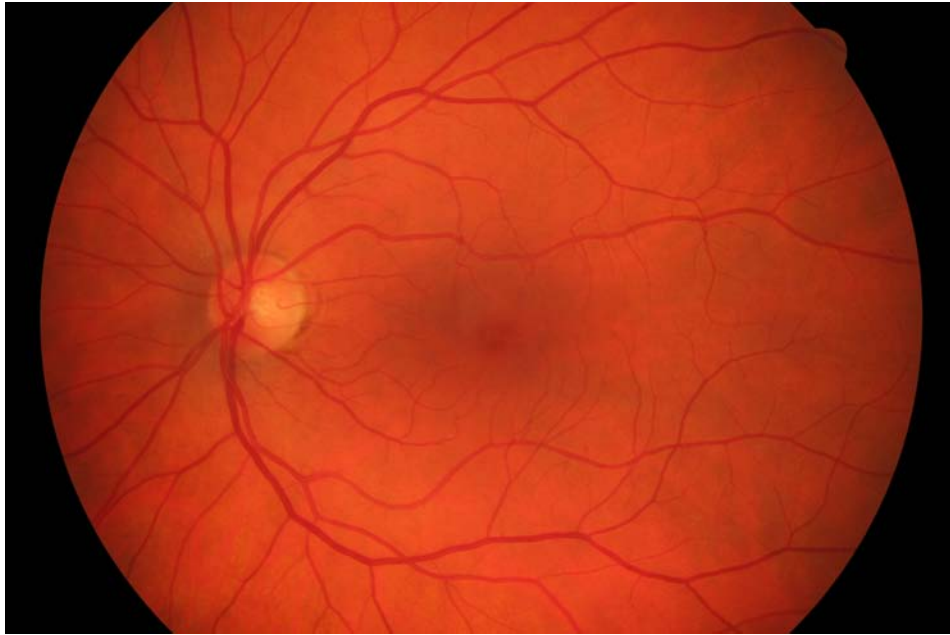


Figure 3.1: Example image of HRF database

Another aspect of this database is that the gold standard for segmentation is manually drawn using a raster graphics editor (ADOBE Photoshop CS4). Some segmented images present imperfections, such as non physiological shape of the vessel tree and serrated edges. Sometimes background pixel around the vessels are segmented as vessels. Despite all of this imperfections, the quality of the segmentation is quite good and in this work is always taken as ground truth. An example image taken from HRF database can be found in figure 3.1.

3.1.2 VAMPIRE software suite

VAMPIRE (Vascular Assessment and Measurement Platform for Images of the REtina) is a software application for efficient, semi-automatic quantification of retinal vessel properties with large collections of fundus camera images. Most of the processing is performed automatically before user intervention, which is kept at minimum. It is an international collaborative project of several image processing groups and clinical centres in Europe, Asia and the UK. The system aims to provide efficient and reliable detection of retinal landmarks (optic disc, retinal zones, main vasculature), and to quantify key parameters frequently used in investigative studies, currently vessel width, vessel branching coefficients, and tortuosity. Vampire software consists of two main tools: VAMPIRE and VAMPIRE-Annotation Tool. During the development of this work and especially for the validation of

the resulting synthetic images, both softwares have been very useful. VAMPIRE was born in 2011, but is in continuous updating and development. With the VAMPIRE software it is possible to process sets of different size and resolutions of retinal fundus images. A user-friendly MatLab[®] interface is provided with which the user can have a visual feedback of the features extracted and choose between a set of tools that allow to easily identify, locate and correct wrong measurements. The identification of the OD can be done both in manual or automatic way. The algorithm used for the automatic identification of the OD is the one presented by Giacchetti et. al. [12]. The automatic segmentation of the vessels is done using a version of the well-known Soares et al. technique proposed in [35]. The algorithm applies a multi-scale, 2-D Gabor wavelet transforms to emphasize the appearance of vessels, followed by supervised pixel classification with a Bayesian classifier. The images are then classified pixel by pixel in two classes, *vessel* or *non-vessel*. A tree-like representation of the vasculature, computed with this algorithm, is used as a preprocessing for measurements. In our work we used the VAMPIRE segmentation for quantitative validation of the images obtained; details on the quantitative tests can be found in chapter 4. VAMPIRE software also allows to determine accurately in a semi-automatic way the width of vessels in specific locations chosen by the user and to estimate other features of the retinal images, like vessel tortuosity and branching coefficient. VAMPIRE-Annotation Tool has been developed to acquire ground truth to validate the algorithm. With the annotation tool, a user can manually segment and locate features on the retinal images e.g. OD, vessel calibre, vessel segmentation and macula. An example of usage of the Annotation tool can be found in section 3.3.9.

3.2 Description Of The Method

We face the problem of creating a realistic synthetic vasculature by dividing the algorithm in two main steps. The first step is the creation of the vessel tree, that outlines all of the geometric characteristics, like vessel calibres, angles, tortuosity and lengths. For this step we use a model-based approach, to model the most important features. In the second step, the texture of the vessels is added on the tree, using a patch based algorithm. In the following sections we will give a detailed description of the two steps used and of some arrangements that were necessary for the algorithm implementation and to set the models parameters.

Our purpose was to give a first and simple description of a highly complex system. The algorithm uses therefore, in many cases, naive or simple approaches, e.g. it only considers some characteristics and details. We expect it to be improved in the future, both in the complexity and in the

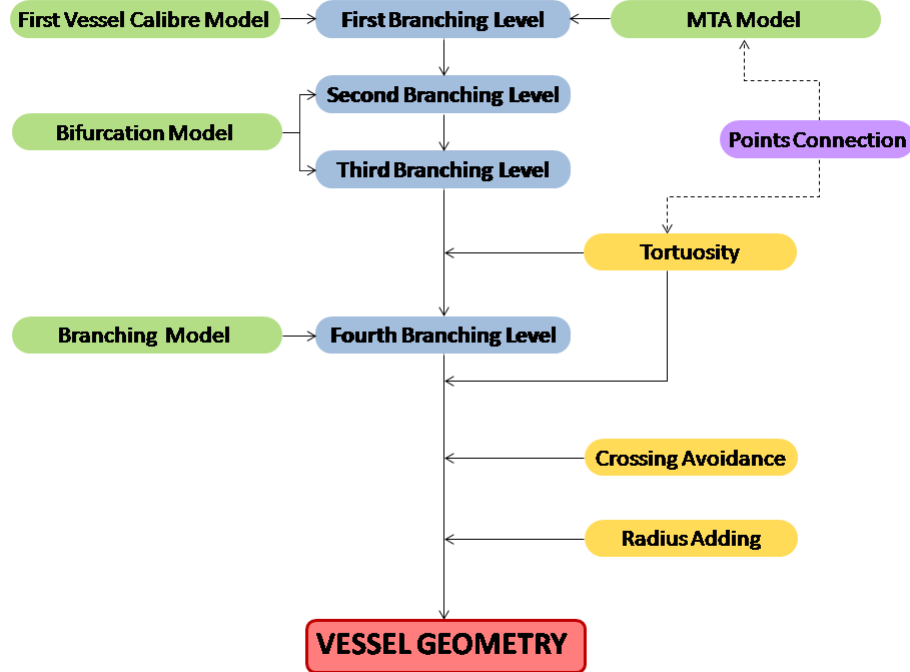


Figure 3.2: Vessel tree geometry synthesis block diagram

features.

3.3 Vessel Geometry

To reproduce the vasculature geometry, we create two separate vessel trees: one for arteries and one for veins. The two different trees are created with the same methodology and models, but the parameters used, in some cases, are different, in accordance to the reproduced vessel type. The setting of those parameters will be discussed later on. The following paragraphs give a detailed description of the methodology used to reproduce one of the vessel trees, the other being the same.

Looking at real images, we can observe that every tree of arteries or veins, in most cases, could be divided into four sub-trees in accordance of the four quadrant in the polar reference system of the retina. In general, the sub-trees all have the same characteristics. The base of our method is then the creation of one of those; an entire tree of arteries or veins will then be formed by assembling four sub-trees with the right orientation and some constraints. For this reason, the sub-tree will be from now on referred to as

tree.

First we decided to divide the generations of the branches in the tree in four branching levels. Each generation represents a specific feature of the entire tree as follow:

- **first level:** is the generation of the branches which follow the Major Temporal Arcade (MTA)
- **second level:** the branches follow a curve that bifurcate from the first level and, depending on the local configuration, diverge or approach the center of the image, in which there is the fovea region.
- **third level:** the branches of the third level complete the second branching generation, in accordance to the bifurcation model.
- **fourth level:** is the generation of the branches that can split from each one of the previous levels with a branching angle of approximately ninety degrees. Usually their calibre is much smaller and so is their length.

The term *branch* is referred to the vessel segment included between two successive bifurcations, or the final segment of the vessel after the last bifurcation or branching point and with respect to *branch length* we also use the summed length along the vessel midline.

3.3.1 First branching level

As we have already introduced before, this tree branching level follows the shape of the Major Temporal Arcade. A collection of points (x, y) are used to represent the MTA, in accordance to the model presented in the section 2.2 and to the shape of HRF database. The first branching level gives the main shape to the vessel tree.

$$(y - y_0)^2 = 4a(x - x_0) \quad (3.0)$$

In equation 3.3.1 the parameter a is randomly chosen between the range $[90, 110]$ for arteries and $[110, 120]$ for veins, in relative accordance to the values found in literature [25], assuming the pdf to be uniform inside the range. Point (x_0, y_0) is the vertex of the parabola, where the vessel tree departs.

Once EP_0 is set as $EP_0 = (x_0, y_0)$ as the first point of the first branch, and $(x_1, y_1) \leftarrow (x_0, y_0)$, the other branches are decided with the following algorithm:

- i. $IP_i \leftarrow EP_{i-1}$
- ii. computes the length of a branch using a gaussian pdf with parameters (μ_1, σ_1)

- iii. from the point (x_i, y_i) calculate the length on the curve using the Freeman distance and arrive to the point (x_{i+1}, y_{i+1}) .
- iv. compute the distance L from the curve using a gaussian pdf with parameters (μ_2, σ_2)
- v. calculate the angle θ of the tangent in (x_{i+1}, y_{i+1})
- vi. $EP_i = (x_{i+1} + L * \sin(\theta), y_{i+1} + L * \cos(\theta))$
- vii. $i \leftarrow i + 1$.
- viii. stop: when the end of the MTA curve is reached.

Details on Freeman distance can be found in section 3.3.8.

Table 3.1: Parameters for the gaussian PDFs, first level

VesselType	μ_1	σ_1	μ_2	σ_2
Artery	600	150	0	80
Vein	650	150	0	80

IP_i and EP_i are respectively the initial and the end point of the i -th branch. The parameters (μ_1, σ_1) and (μ_2, σ_2) are decided by visual inspection and are different between arteries and veins. The parameters are shown in Table 3.1.

It is important to highlight that the MTA model in literature is only representative of the part of the vessel tree close to the OD and can be considered valid approximatively until the region of the fovea. When moving farther away from the OD, the main shape of the vessel tree cannot be correctly modeled though a parabola anymore. In spite of this, we maintain the parabolic model all along the curve; future work on this topic will need to consider this and develop a new model for the farther regions.

Another problem of the model used for the MTA is that the values for the opening parameter a are the same in [25] for both the temporal and nasal arcades in the eye, using the fundus retinal images fovea centered. For our purposes those values are used in the ranges written before for the trees in the upper right and lower right quadrants, but a different range is chosen for a in the quadrants in which are present the nasal arcades. More specifically, the values for these two quadrants are chosen higher, considering the curvature of the retina (that is transposed to a more open arcade on a 2D image).

3.3.2 Second branching level

Every time we set a bifurcation point in the first branching level, a second branching level starts from that point. The i -th bifurcation point in the first branching level is set between the branch i -th and $(i+1)$ -th and is equal to EP_i (or IP_{i+1}). There are $N_1 - 1$ bifurcation points in the first level, where N_1 is the number of branches in the first level. Generally the second branching level is constructed the same way of the that level, with some adaptations.

Consider now the i -th branching point, we will refer to the branch i -th as *parent* and to the branch $(i+1)$ -th as *son*. Referring to the bifurcation model presented in section 2.3, we first need to decide if the son branch is a right son or a left son of the parent. The angle ϕ_i of the parent referred to the cartesian reference system is computed as:

$$\phi_i = \arctan \frac{EP_{iy} - IP_{iy}}{EP_{ix} - IP_{ix}} \quad (3.0)$$

The angle of the son, ϕ_{i+1} , is also calculated with equation 3.3.2. The relative angle θ is given by:

$$\theta = \phi_i - \phi_{i+1}$$

If $\theta \geq 0$ then the son is labelled as a *right* son and $\theta_1 = \theta$, if $\theta < 0$ then the son is labelled as a *left* son and $\theta_2 = \theta$.

The second step requires to compute the vessel calibres. The model that allows us to compute the calibre of the vessels in the first branch of the vessel tree will be discussed later on in section 3.3.9. All other calibres are iteratively computed as follows. At the i -th iteration, known parameters in the bifurcation model are:

- parent calibre $2r_0 = 2r_i$
- son angle θ_1 or θ_2 , computed as before.

The parameters to calculate are:

- left son calibre $2r_2$
- right son calibre $2r_1$
- left son relative angle θ_2

Let us assume θ_1 is known (if θ_2 is known we just need to invert θ_1 and θ_2 in the formulas); the other parameters in the bifurcation model are computed with equations 2.3, 2.3, 2.3.

In equation 2.3 r_1 is computed finding the zero of the function:

$$f(r_1) = (r_0^4 + r_1^4 - (r_0^3 - r_1^3)^{\frac{4}{3}}) - 2r_0^2 r_1^2 \cos \theta_1$$

The parameters θ_2 , r_1 and r_2 are then multiplied by three different coefficients. Each coefficient is determined randomly from a uniform distribution between 0,8 and 1,2.

If the (i+1)-th branch is a right son, then we assign to the branch (i+1)-th the calibre $r_{i+1} = r_1$. To the i-th branch we assign a left son with calibre $2r_2$ and relative angle θ_2 .

If the (i+1)-th branch is a left son, then we assign to the branch (i+1)-th the calibre $r_{i+1} = r_2$. To the i-th branch we assign a left son with calibre $2r_1$ and relative angle θ_1 .

In each bifurcation point the remaining son just computed is the first branch of the second level. According to its orientation with respect to the parent branch, there are two possibilities now:

- the branch is a left son: the second branching level has to diverge from the image center (fovea region);
- the branch is a right son: the second branching level has to approach the image center.

If we refer to the son just computed as the J-th branch, where J is plus one the number of branches already computed ($N_1 +$ number of second level branches already computed at the i-th iteration), the other second level branches at the i-th branching point can be computed with the same algorithm used for the first branching level with $j = J+1, J+2, \dots$

Once again, the model assumes that the branches follow a parabolic shaped curve, with vertex in EP_i and rotated so that it is converging to or diverging from the image center for right or left son branches respectively.

Another constraint is that the first branch of the i-th second branching level is the son J-th, so its orientation is already computed. $IP_J = EP_i$ and EP_J is calculated using the formula:

$$EP_J = (IP_{Jx} + D \cos(\theta + \phi_i), IP_{Jy} + D \sin(\theta + \phi_i)). \quad (3.0)$$

D is the length of the branch, that is taken from a gaussian distribution. The point (x_{J+1}, y_{J+1}) on the curve is then given and computed calculating the point on the curve that has the minimum distance from EP_J .

The averages and standard deviations for the gaussian distributions used for the branch lengths and the distances from the curve in the second level can be found in table 3.2.

3.3.3 Third branching Level

After adding the second branching level we have $N_2 - (N_1 - 1)$ branches with a single son. The third branching level is used to complete the bifurcations

Table 3.2: Parameters for the gaussian PDFs, second level

VesselType	μ_1	σ_1	μ_2	σ_2
Artery	600	150	0	40
Vein	650	150	0	40

remained; notice that the bifurcations starting from the first level are already complete. With the same algorithm used for the second branching level at each bifurcation point with a single son, the other son branch is added. The distance between IP_k and EP_k , with k meaning branches of the third level, is computed with with a shape different from the Gaussian that was used so far. Among the various pdf shapes that were considered in order to get more realistic images, a uniform pdf is the one that returns the best results to complete the second level bifurcation with the remaining branches. The distances were chosen in the range $[200, 300]$ for both arteries and veins.

3.3.4 Tortuosity adding

All the IP_i and the EP_i points of the branches from the first, second and third level have now been initialized. We now need to define the centerline of the branches. In most of the previous works discussed in section 2.1, the branch is represented by a simple line linking two successive bifurcations. To make our model more realistic, we introduce instead some kind of tortuosity on the vessels. A tortuous vessel can be defined as a vessel that twists and winds in a "tortuous" path. There are no specific definitions of tortuosity, it is therefore impossible to give a unique quantitative definition of this concept. To our best knowledge there are not in literature complete models of a tortuous vessel but only a wide range of methods based on different kind of measurement of some "curvature" properties of the vessel.

In the current work, we add the centerline of each branch using a spline. We choose to use a spline because usually the variations in curvature are smoothed along the centerline of a vessel and with a spline we can control in a very precise way the points in which the curvature changes.

For every branch i -th in the tree we take a segment on the x -axis in cartesian coordinates of the same length L between IP_i and EP_i ,

$$\{x \in \mathbb{R} \mid x \in [0, L]\}$$

On this segment we define a collection \bar{x} as

$$\bar{x} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$$

where n is the number of control points on the spline and $\bar{x}_1 < \bar{x}_2 < \dots < \bar{x}_n$. The first and the last point on the spline as to be equal to IP_i and

EP_i respectively, this fact is reached imposing that $\bar{x}_1 > 0$ and $\bar{x}_n < L$. The collection of control points of the spline is defined by (\bar{x}_j, \bar{y}_j) , where $\{\bar{x}_j\}_{j=1,2,\dots,n} = \bar{x}$ and the collection \bar{y} is setted with the following algorithm:

- I. control if the i -th branch is a *left* or a *right* son
- II. if branch i -th is a *right* son a parameter k_1 is set as 1
- III. if branch i -th is a *left* son, the parameter k_1 is set as -1
- IV. if the branch is the first branch of the tree, so it is not classified as a *right* or a *left* son, the parameter k_1 is taken randomly between 1 or -1.
- V. proceed iteratively for $j = 1, 2, \dots, n$
 - i. compute a distance d using a gaussian pdf with parameters μ_y and σ_y
 - ii. $y_j = k_j|d|$
 - iii. $k_{j+1} \leftarrow -k_j$

The distances between the $\{\bar{x}_j\}_{j=1,2,\dots,n}$ points were also decided using a gaussian pdf, with parameters μ_x and σ_x . The parameters used for both the gaussian pdf can be found in table 3.3.

Table 3.3: Parameters for the gaussian PDFs, tortuosity

μ_x	σ_x	μ_y	σ_y	level
150	20	0	10	1
150	20	0	10	2
150	10	0	10	3
60	5	0	7	4

The points $(\bar{x}_j, \bar{y}_j)_{j=1,2,\dots,n}$ are then connected using a cubic spline finding the curve $y(x)$ with $x \in [0, l]$.

The curve $y(x)$ is rotated by an angle α , which is the angle of the segment between IP_i and EP_i , using the formula 3.3.2. To rotate $y(x)$ of an angle α , we switch from cartesian to polar coordinates and add the angle α to the phase. We return then to cartesian coordinates. The curve $y_r(x)$ is then translated in the point IP_i . We can now define two parametric curves $x(t)$ and $y(t)$ with $t \in [0, l]$ from $y_{rt}(x)$ that determine the centerline of the i -th branch.

Using this algorithm to add a tortuous vessel and changing the parameters of the gaussian pdf, many levels of tortuosity can be reached . For

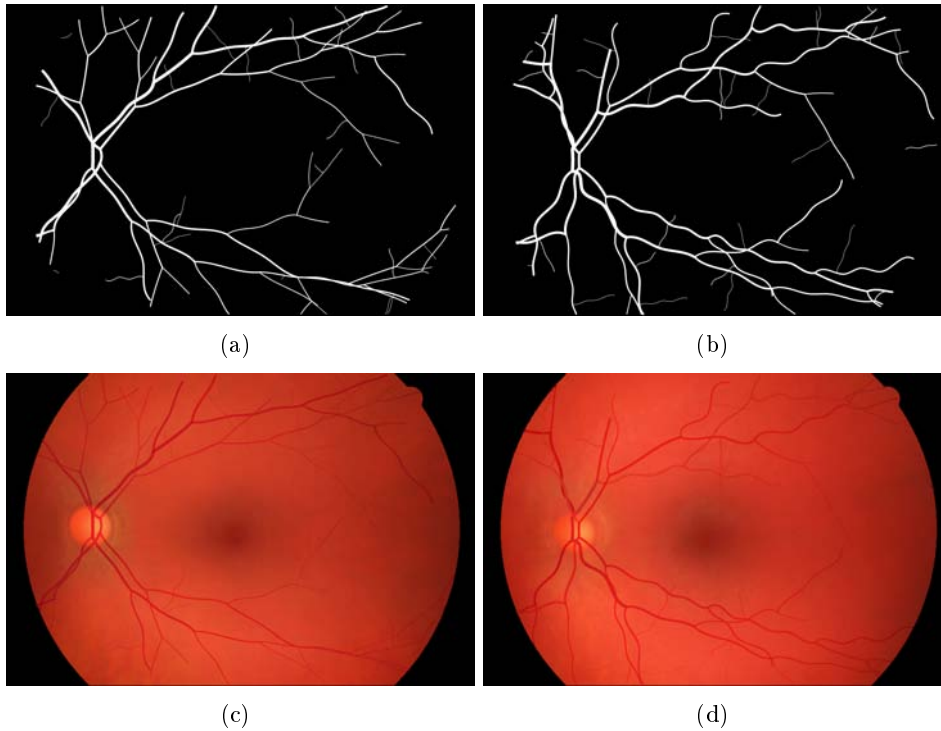


Figure 3.3: Examples of different levels of tortuosity: (a)(c) less tortuous vasculature, mask and relative synthetic image; (b)(d) more tortuous vasculature, mask and relative synthetic image

example the parameter σ_y regulates the linearity of the curves, i.e. if they are more or less accentuated. the parameters μ_x and σ_x , instead, are responsible for the number of curves.

In order to count the number of changes in concavity along the vessel midline, we add a control on the spline control points. This control wasn't necessary for this work, but it is very useful for validation of algorithms that calculate tortuosity on the base of concavity changes of a curve, or that combine it with other parameters [19].

3.3.5 Fourth branching Level

As we have mentioned before, the fourth level is the one that realizes the branching. With the term branching we mean those bifurcations in which a son is much smaller than the parent and the other son, the relative bifurcation angle of the smallest branch is around ninety degree, and the relative angle of the largest son is close to zero. The bifurcation model fails in describing this type of bifurcation. A branching can be also described as a vessel that departs from the parent branch at a random point along the midline of the

parent branch, without changing the parent branch orientation. The calibre of the branching vessel is too small to alterate the flux of the parent branch in a conspicuous way, thus we suppose that the vessel calibre of the parent branch remains unchanged after the branching point. To add branches to the tree constructed with the first tree branching levels, On the base or real images, we must take into account that a vessel might have one branching point, more than one branching point, or no branches at all.

We start from the first branch and proceed iteratively along the tree we just built.

- i. at i -th branch take a random number n between 0, 1 or 2. This is the number of the branching point to add on the i -th branch.
- ii. if $n > 0$ then on the i -th branch centerline compute n branching point/s (x_p, y_p) .
- iii. calculate the tangent in the (x_p, y_p) point.
- iv. calculate the perpendicular angle β in (x_p, y_p) point to the curve.
- v. add variation on the angle β .
- vi. set $IP_f = (x_p, y_p)$. This is the starting point of a new fourth level branch.
- vii. compute the distance d and calculate the EP_f with the following formulas:

$$EP_{fx} = IP_{fx} + d \cos \beta$$

$$EP_{fy} = IP_{fy} + d \sin \beta$$
- viii. if n is equal to 2 then compute again steps from iii. to vii.
- ix. $i \leftarrow i + 1$.

The number f , given as a reference to the branch, has to start plus one the number of branches already computed in the first three levels.

We have choosen to add maximum 2 branches on a vessel, looking at the images we obtained in an empirical way. Further works could include a statistics on the number of branching points along the vessels. The value d of the distance between IP_f and EP_f is taken from a uniform distribution in the range [200, 400] for both arteries and veins. To add variability also in the direction of branching (to the left side or to the righth side of the parent branch) the sign of the distance d is choosen randomly.

To compute the perpendicular angle β to the curve in each point (x_p, y_p) we could we could follow two strategies.

- A first approach is to consider the information we acquire from the use of a spline to connect the initial and ending point of the branches; since we know the parameters of the spline, we are able to calculate the perpendicular direction in each and every point of the curve.
- The second strategy was to treat the centerline of the branch as an unknown curve and to calculate the angle β approximating the direction of the curve in the (x_p, y_p) point.

Although the first way seems to be more precise and correct, some observation made us choose the second way. As it will be shown in the next section, due to the implementation with MatLab[®] of the spline curve and the discretized level of the problem, it might happen that the function is not defined in a certain point we are interested in. So we had to add a step in our algorithm to control the connection of the points that will be described next. Moreover, there is no need of high precision in the branching angle and the first method requires a higher computational effort; for future implementation it could be that our way to add tortuosity would change. Finding an efficient implementation of the first approach could be part of future work on this subject.

To compute the perpendicular angle β we first take a subset of the points on the curve around (x_p, y_p) . The width of this neighborhood is an empirical parameter. The best regression line is then fit to this collection of points. The value of the angle β is calculated as:

$$\beta = \arctan\left(-\frac{1}{m}\right)$$

where m is the angular coefficient of the regression line. The angle β is then modified by multiplying it by a coefficient taken from a uniform range $[0.9, 1.1]$ in a similar way to what we did for the bifurcation parameters. This is due to the fact that branching angles are almost ninety degrees, but in real branches some variation are possible.

To avoid the growth of a fourth level branch too close to an already computed bifurcation point, we have imposed some boundary conditions. To compute the branching points (x_p, y_p) we proceed as described for the point search algorithm on the MTA curve in first branching level section. We then need Freeman distance for this step as well, and we calculate it along the parent branch centerline

After having initialized all the IP_f and EP_f of the fourth level, we perform the tortuosity step once again to get the centerline of the branches of this last level. The parameter used for the centerlines at this step can be found in table 3.3. Such parameters are different from those of the previous levels; this is due to the fact that branches are now smaller and shorter than

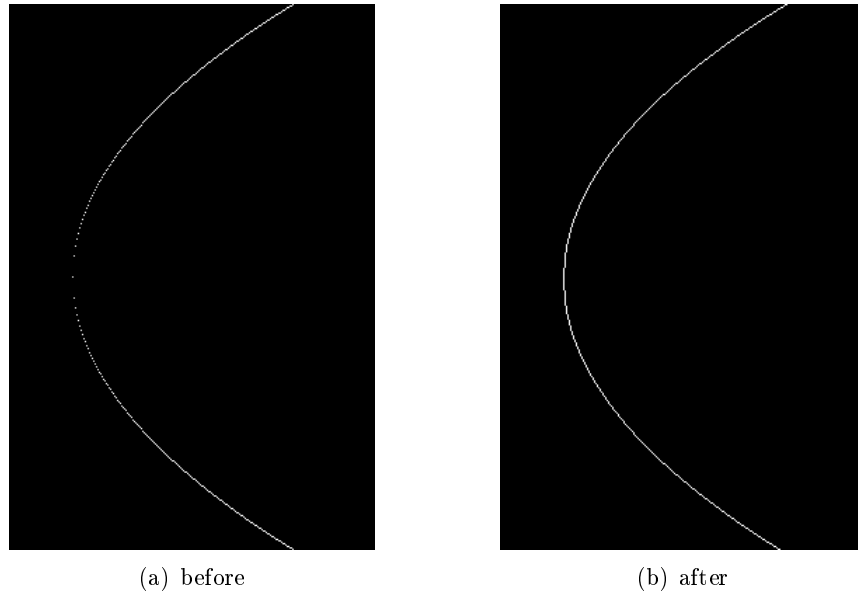


Figure 3.4: MTA curve example before and after point connection step

before, and this generally results in more tortuous vessels

To create an entire vasculature system four branches of the same type have to be initialised. As we have mentioned before our algorithm allows to create a tree only for the first quadrant. To create the vasculature system the of the three remaining sub-trees, the described system has to be reflected with respect to the axis passing through the OD. The resulting structure is then checked, to eliminate branches or part of branches that cross with vessels of the same type.

3.3.6 Curve point connection

In the previous paragraph we mentioned a problem deriving from the discretization of the images. Passing from continuous formulation of functions to their discretized formulation, a curve can have then some unconnected points. To compute the various steps of our algorithm it is necessary that all the curves are connected, i.e. the curves used to create the various branching levels as well as the branches centerlines. For example using the formula 3.3.1 to create parabolas, especially in the first part of the parabola (close to the vertex), it happens really very often that the points are unconnected. See figure 3.4 for explanations. This could be avoided taking a higher sampling rate of the parametric curves $y(t)$ and $x(t)$, but in most of cases a higher sampling rate could decrease the performances in term of time of the algorithm and would be necessary again another step in order to avoid

double sampling in discretized curves. We therefore decided to take a lower sampling rate and to add a step to join all the points on a curve.

To be connected, all the consecutive points on a curve have to be *8-adjacent*. We define three different types of neighborhood of the a pixel p of coordinates (\bar{x}, \bar{y}) :

- $N_4(p)$ **4-neighbors** : $(\bar{x} + 1, \bar{y}), (\bar{x} - 1, \bar{y}), (\bar{x}, \bar{y} - 1), (\bar{x}, \bar{y} + 1)$
- $N_D(p)$ **D-neighbors** : $(\bar{x} + 1, \bar{y} + 1), (\bar{x} + 1, \bar{y} - 1), (\bar{x} - 1, \bar{y} + 1), (\bar{x} - 1, \bar{y} - 1)$
- $N_8(p)$ **8-neighbors** : 4 - neighbors and D-neighbors

We define the *8-adjacency* between two pixels p and q if :

- they have values in the same set (for a binary image if they are both white or both black);
- q is in $N_8(p)$

Two pixels are then connected if they are *8-adjacent*.

Suppose that a discretized curve is defined by a collection of points (\bar{x}_i, \bar{y}_i) , with $i = 1, 2, \dots, N_c - 1$ where N_c is the number of points of the curve. To connect all the points we proceed iteratively from the first to the last point:

- i. start from $i = 1$;
- ii. if (\bar{x}_i, \bar{y}_i) and $(\bar{x}_{i+1}, \bar{y}_{i+1})$ are connected then $i \leftarrow i + 1$
- iii. if not, then calculate the linear approximation between them and add the pixel of this line in the collection of pixel that defines the curve, then $i \leftarrow i + 1$.

After the connection has been performed, one more step is needed to skeletonize the curve, i.e. to eliminate superfluous points in order to have at most two pixels in each 8-neighbourhood (see section 3.4: *Vessel Calibre Mask* for further details). See figure 3.5 for details.

3.3.7 Radius adding

For the creation of the binary masks, that defines the segmentation of the vasculature from the tree structures just built, it was necessary to implement a method to add the vessel width along the centerlines computed in the tortuosity step after the definition of the four-level branches tree. The next step of the algorithm is the creation of a binary mask of the entire vascular system. Up to now, we have created the centerlines of the vessels and we have made sure that they are continuous. We now need to consider the entire

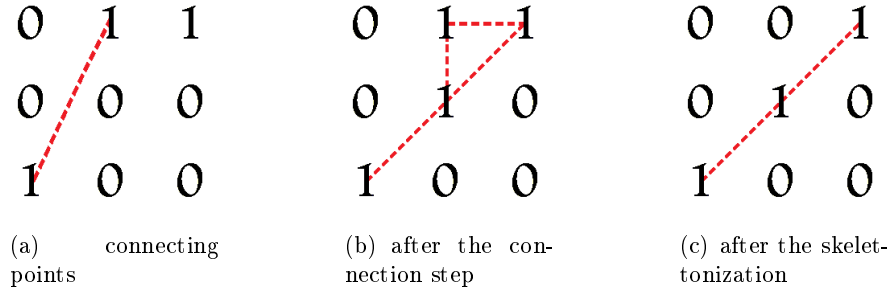


Figure 3.5: Connection and skeletonization of pixels on a curve

vessel around a centerline. Since we have a structure that we want to follow along the all image, a possible approach it to use morphological operations on such structure. If on a binary image we set the centerlines pixels to "1" and all the other pixels to "0", the idea is then to compute the morphological operator of **dilation** to the image in order to obtain the binary mask. A definition of the morphological operation of **dilation** is the following: Given two binary images A and B (A and $B \in \mathbf{Z}^2$) the dilation of A with B is

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

where

- $R = \hat{B} = \{y | y = -x, x \in B\}$ is the reflection done on the B image
- $(R)_z = \{y | y = x + z, x \in A\}$ is the translation of R with a vector z .

B is called the *structuring element* and usually its dimensions are much smaller than A dimensions. The **dilation** is then the set of the points z which, when \hat{B} is translated in those points, at least a point in \hat{B} is included in A . There are many other definitions of dilation, like the one that uses the Minkowski addition

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

All the definitions of the morphological operation of dilation are global operation on the target image. Since every branch has its own calibre, we need to define a local opening operation along the all structure. For this purpose, we define the *structuring element* B as the approximation of a disk with radius r related to the calibre of the considered branch:

$$r = \frac{\text{calibre}}{2} \quad (3.0)$$

MatLab[®] provides a function to create the *structuring elements*, called `strel`, with whom can be created elements of many different forms. We

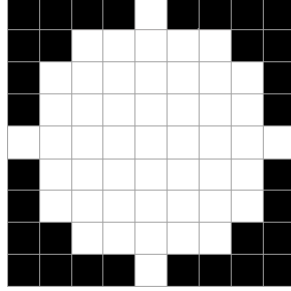


Figure 3.6: Example of structuring element of radius 4

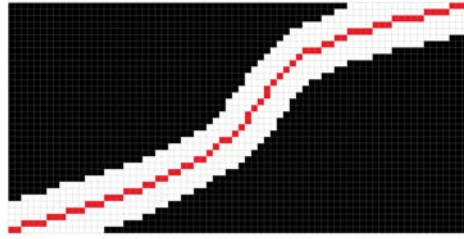


Figure 3.7: Dilation with a disk of radius 4 applied to a centerline (red)

choose the *disk* form (figure 3.6) because the resulting edges of the binary masks are homogeneous and smooth.

To add the calibre on a branch the following steps are applied:

- i. At branch i -th, create the structuring element B with the appropriate radius (equation 3.3.7)
- ii. For every pixel \in centerline calculate the dilation using the structure element B
- iii. Add the branch dilate on the binary mask.
- iv. $i \leftarrow i + 1$

3.3.8 Freeman distance

With the term *Freeman distance* we referred to the distance between two points on a curve calculated according to the Freeman chain code theory [10].

The Freeman chain code is a compact way to represent the contour of an object. The chain code is an ordered sequence of n links $\{c_i, i = 1, 2, \dots, n\}$, where c_i is a vector connecting neighboring edge pixels. The directions of c_i

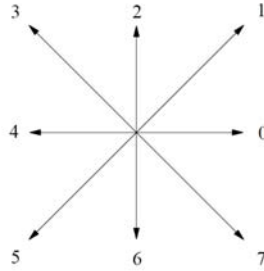


Figure 3.8: The directions of the eighth connected Freeman chain code ($K = 8$)

are coded with integer values $k = 0, 1, \dots, K - 1$ in a counterclockwise sense starting from the direction of the positive x-axis (see figure 3.8). The number of directions K takes integer values $8M$ where M is an integer. The chain codes where $K > 8$ are called the generalized chain codes. Suppose to have a black and white image, a curve on this image can be represented by two parametric curves $x(t)$ and $y(t)$, where $t = 1, 2, \dots, N_p$ with N_p is the number of pixel in the image that belongs to the curve. We took $K = 8$. Starting from the first point on the curve $(x(1), y(1))$ we can compute the length on the curve iteratively, considering the direction between consecutive points.

- i. initialize length $l \leftarrow 0$
- ii. compute the direction d between $(x(i), y(i))$ and $(x(i + 1), y(i + 1))$ $i = 1, 2, \dots, N_p - 1$
- iii. if d is equal to 0 or 2 or 4 or 6 then $l \leftarrow l + 1$
- iv. if d is equal to 1 or 3 or 5 or 7 then $l \leftarrow l + \sqrt{2}$

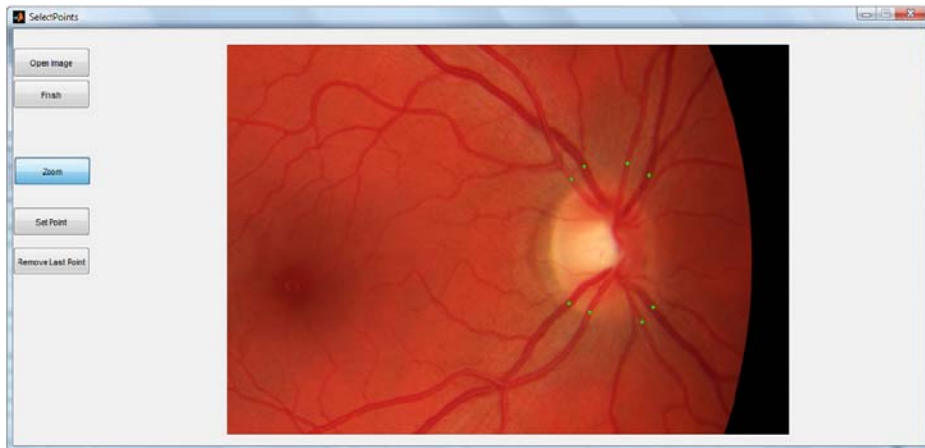
This method is often used throughout this work, when it was necessary to calculate the distance on a curve between two distinct points.

3.3.9 First vessel calibre model

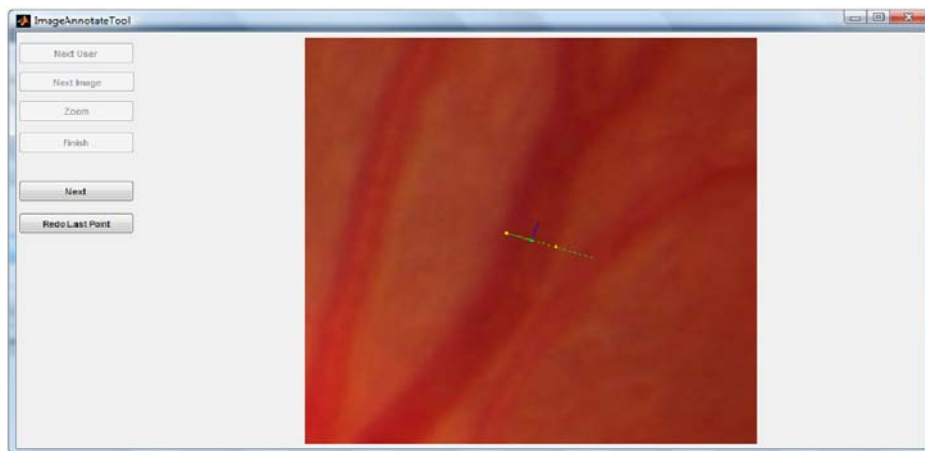
As we mentioned before, when we described the way the first branching level is realized, it is necessary to have a model for the calibre of the first vessel of the tree. The other calibres are then computed using the bifurcation model. The calibre of the first vessel is important for the overall aspect of the vessel tree, because all the other calibres are derived from it and cannot be wider than it.

To compute a distribution for the calibres we collected data using VAMPIRE Annotation Tool ²[30] for vessel calibres. In figure 3.9(a) the points in

²free to download: <http://vampire.computing.dundee.ac.uk/tools.html>



(a) selection of points



(b) manual annotation

Figure 3.9: Example of Annotation Tool usage

which we annotate the various widths (figure 3.9(b)) are taken on the vessels around the optic disk. We needed data from the first branch of the tree, that can be also seen as the first branch of each arcade; however, it is sometimes difficult to identify such branch, since the bifurcation might happen before leaving the OD. In these cases, we choose to collect the data from the first widest branch around the OD.

From each of the fifteen real image HRF healthy database we collect widths from four arteries and four veins, resulting in a total of 60 artery and 60 vein widths. The sample mean and standard deviations of this datasets can be found in table 3.4.

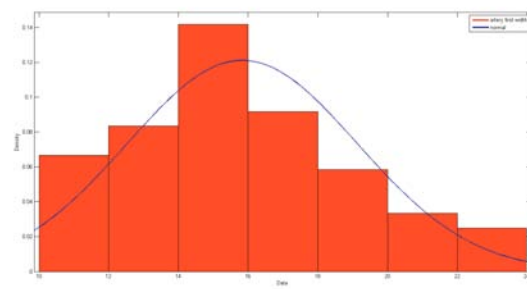
Table 3.4: Sample mean and standard deviation of first vessel calibre data

Vessel Type	<i>mean</i>	<i>sd</i>
artery	15.83	3.29
vein	21.08	4.86

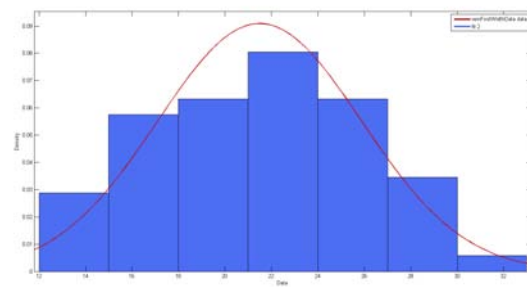
To model this collection of data we took advantage of the MatLab[®] Distribution Fitting app. This useful MatLab[®] App allows the user to fit the data with many well known distributions, e.g. Normal, Log-Normal, Logistic, Log-Logistic, Gamma etc. or to customize a distribution. The curve distribution can be displayed upon the normalized histogram of the data. This App also allows to set the exclusion rules for the outliers and the bin dimension. In all our fittings we used the Freeman-Diaconis rule for the setting of bin dimension.

$$Bin\ size = 2IQR(x)n^{-\frac{1}{3}}$$

where $IQR(x)$ is the interquartile range of the data and n is the number of observations in the sample x . The distribution fitted and the normalized histograms of the arteries and veins first vessel calibre can be found in figure 3.10. The parameters of the fitting can be found in table 3.5. Among all the different distributions, we choose a Normal. However we had to add again some boundary conditions to the model to avoid too small or too wide first branches. The bounds for arteries are [10,24], for veins [12, 32], taken from the minimum and maximum values found in HRF database. We can see, as we expected, that the calibre of veins is generally larger than the calibre of arteries. The values obtained as parameters for gaussian distributions do not vary significantly from the sample average and standard deviations. This fact and the visual inspection of the histograms allowed us to think that the gaussian pdf was the one that most suited for the models.



(a) arteries



(b) veins

Figure 3.10: Gaussian distribution fit and histograms of first vessel calibre datasets

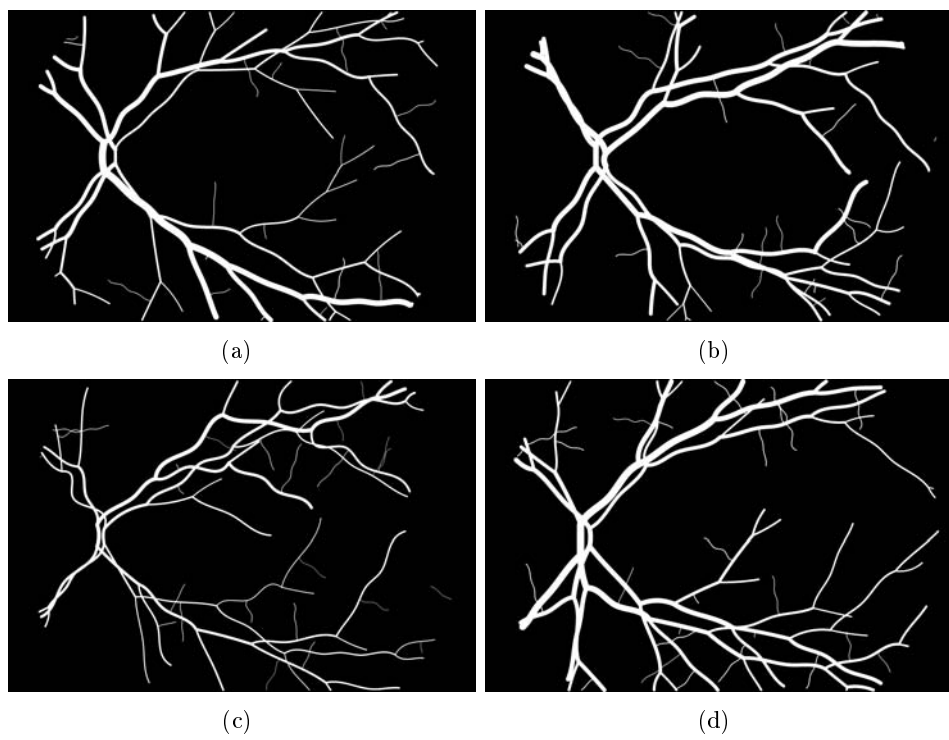


Figure 3.11: Examples of vasculature masks created with the proposed algorithm

Table 3.5: Parameters for the gaussian PDFs, first vessel calibre

Vessel Type	<i>mean</i>	std. err.	<i>sd</i>	std. err.
artery	15.83	0.42	3.30	0.31
vein	21.49	0.58	4.39	0.41

3.4 Vessel Texture

A second algorithm has been implemented to add the texture on the vessel tree. From the Vessel Geometry algorithm we have defined a structure that contains all the necessary informations to construct the tree, such as the positions of the branches, the centerlines and the radii. To add the texture on the vessels we use a patch-based algorithm based on the well known Quilting technique [6]. Quilting is a method of generating novel visual appearance in which a new image is synthesized by stitching together small patches of existing images. No further details will be given about this technique and on the implementation procedure it has been implemented because it is part of a parallel work on synthetic background and optic disk of retinal images [7]. In [7] a detailed explanation can be found on the methods and solutions used to implement quilting for retinal images. We will dwell only on details that differ from Fiorini's work and on the solutions found. The idea is to apply the quilting technique to build a linear vessel of length equal to the branch centerline length. The next step is to warp the linear vessel, to follow the centerline of the branch. Then the branch with the texture is hooked with the synthetic background created with [7]. As we have just mentioned, patches of real images are necessary for quilting. The first thing to do is the creation of a Tiles dictionary, used to construct the linear vessel. The quilting technique has the merit of creating a new synthetic image, using data collected from real images. To be as accurate as possible, we divide the tiles dictionary in two main categories, *arterial tiles* and *veinous tiles*. To create the tiles dictionary were used many different masks and maps and a intuitive solution applied on real images to make them more homogeneous.

3.4.1 Mask and maps

Artery and Vein Classification mask

To collect tiles from arteries and veins, we have to know where they are on real images, so we collect them from the 15 healthy images in the HRF database. The HRF database also provides manual segmentation of images, but not the classification of arteries and veins. To obtain the A/V classification we proceed manually using the vessel segmentation in comparison with real images, marking arteries and veins with two different colors. Therefore we obtained 15 mask of arteries and 15 masks of veins. The classification

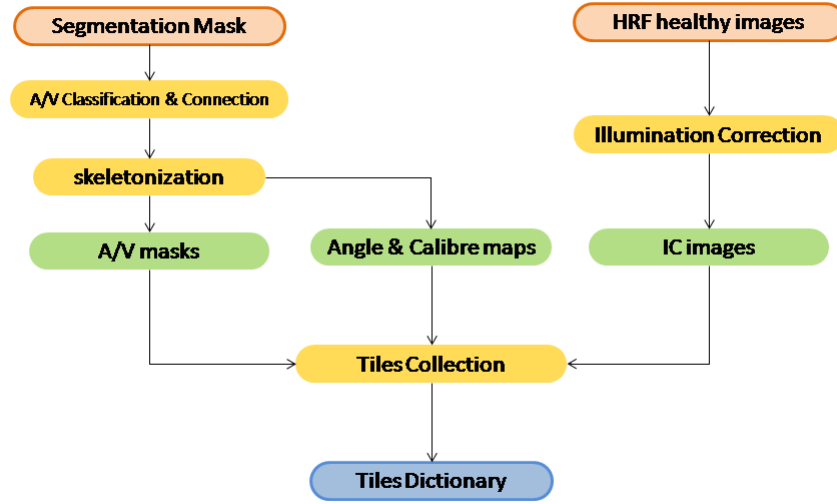


Figure 3.12: Tiles collection block diagram

has been made by a non-clinician user, using the freeware software IrfanView³. Some vessels that are present on the segmented images, but of whom the classification was uncertain, were completely deleted on the new masks created. An example of classification mask can be found in figure 3.13.

From those masks, different segmentation masks for arteries and for veins are then obtained, by applying two different threshold values. A further manual step was done in order to connect all the branches that were separated in the masks due to the crossing points between arteries and veins. (see figure 3.14 for details).

Vessel calibre map

A first type of map that is necessary for tiles collection is the Calibre masks, in which the widths of the vessels in every point are computed and stored. To construct the vessel calibre masks we have to shortly introduce the **distance transform**. The distance transform is an operator normally only applied to binary images. The result of the transform is a graylevel image that looks similar to the input image, except that the graylevel intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point. There are several different sorts of distance transform, depending upon which distance metric is being used to determine the distance between pixels. To our purposes we used the Euclidean metric to compute distance transform. The idea is to apply the distance transform

³in free download at:<http://www.irfanview.com/>



Figure 3.13: Example of A/V classification from segmentation mask

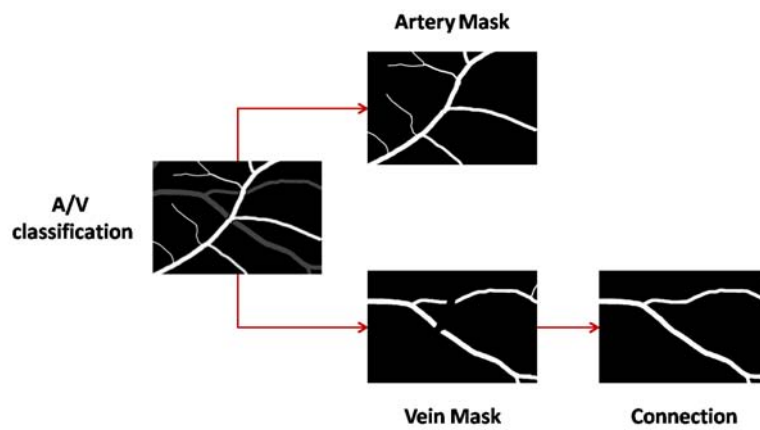


Figure 3.14: Unconnected vein branches. After the separation in arteries and veins masks, the branches are manually connected.

on the binary classification images and to the mask then the centerlines (that defines the distance from the center pixel to the edge of the vessel, so they are the radii along the centerline of each branch). To mask the image obtained with distance transform we need the centerlines of the classification masks. They can be calculated with the skeletonization technique. To obtain the skeleton of an image is necessary to follow these steps [18]:

- I. iterate on every pixel of the image p of coordinates (x_i, y_i) . Define the 8-neighborhood of the pixel p as a collection of pixel $\{p_k\}_{k=0,1,\dots,7}$, where k refer to the neighbor pixel in direction k (see figure 3.8). Delete the pixel p if all the following conditions are satisfied:

- i. Define the crossing number $X_H(p) = \sum_{j=1}^4 b_j$ where

$$b_i = \begin{cases} 1 & \text{if } p_{2j-2} = 0 \text{ and } (p_{2j-1} = 1 \text{ or } p_{2j} = 1) \\ 0 & \text{otherwise} \end{cases}$$

if $X_H(p) = 1$ then the first condition is satisfied

- ii. $2 \leq \min\{n_1(p), n_2(p)\} \leq 3$ where:

$$n_1(p) = \sum_{k=1}^4 p_{2k-2} \text{ or } p_{2k-1}$$

$$n_2(p) = \sum_{k=1}^4 p_{2k-1} \text{ or } p_{2k}$$

- iii. $(p_5 \text{ or } p_7 \text{ or } \neg p_7) \text{ and } p_0 = 0$;

- II. iterate again in every pixel p of the image and erase the pixel if all the conditions i. and ii. of the previous step are again satisfied and also we have $(p_5 \text{ or } P_6 \text{ or } \neg p_3) \text{ and } p_4 = 0$.
- III. Repeat I. and II. until the image stops changing.

To compute the skeleton of an image MatLab[®] provides a collection of functions; the algorithm just exposed is implemented as an instance of the function `bwmorph`, giving a certain set of parameters.

To obtain the Vessel Calibre masks:

- i. compute the skeletonization S_i of each binary mask of connected arteries or veins M_i
- ii. compute $\bar{M}_i = \neg M_i$
- iii. calculate the distance transform D_i on \bar{M}_i
- iv. take only the centerlines and calculate the calibre $C_i = 2 D S$, multiply D and S pixel per pixel.

The masks C are the masks in which every point on the centerlines represents the vessel calibre in that specific point.

Bifurcations and Crossings masks and Angle maps

The last set of masks that was created included the masks for bifurcations and crossings, to avoid tiles collection on such points. Bifurcations masks were created from the skeleton version of the binary A/V masks looking at the 8-neighborhood of each pixel. If it contains more than two white pixels, the pixel p is marked as a bifurcation point. To construct the bifurcation mask a dilation is applied on the threshold mask of bifurcations points.

The masks of the crossing points between arteries and veins are computed in a similar way. Comparing artery and vein skeleton mask that correspond to the same original image, the crossing points are the one that are set to white in both masks. Once again, we apply a thresholding and a morphological dilation, the latest being meant to have security bounds to not collect tiles in bifurcation and crossing points.

The angles maps are built using the skeleton masks. Along the centerlines the angles are computed in a similar way of angle β in *Fourth Branching Level*. Calling m the angular coefficient of the centerline in p , then angle α in p is $\alpha = \arctan m$.

3.4.2 Illumination correction

One of the problems we had to face while using the quilting technique to model the vessels texture was that the quality of the resulting images was lower than expected. The quilting technique can be described as the assembling of a puzzle: each patch should fit exactly to the neighbouring ones. When no corresponding patch is found, the algorithm chooses the most similar, but this results in artifacts. Since the same algorithm applied to the background dealt far better results, we assumed that the problem was connected to the number of available patches: 300,000 for the background, only about 1,000 for the vessels. An additional explanation is the illumination of the vessels. Because of the acquisition process, retinal images are usually non-uniformly illuminated and exhibit local luminosity and contrast variability, so that even vessels that are similar for type and calibre may appear much different depending on their location. Unfortunately, it was not possible to directly solve the first issue by just increasing the number of tiles because this would require more images acquired under the same conditions and with the same post-processing, which are not available. The second issue was instead dealt with follows.

The idea was to make the illumination of all tiles as similar as possible not only for tiles of the same images, but also in different images. Grisan et al.[13] proposed a model-based correction method based on the illumination of retinal images, in which the illumination of the image is estimated as a combination of three-dimensional functions on a sub area. RGB color space is not suited for illumination correction and chromatic conservation, since the

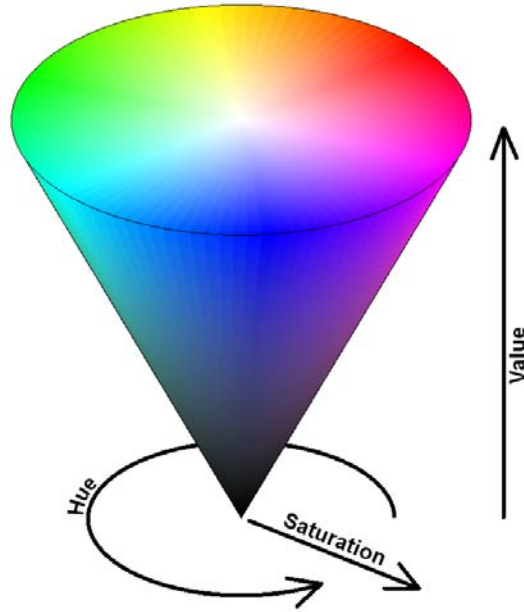


Figure 3.15: HSV (Hue Saturation Value) colour space representation

channels are correlated and therefore do not allow to esteem the perceived light. We therefore choose to use the HSV (hue, saturation, value) colour space, in accordance to [13]. A visual representation of HSV color space is shown in figure 3.15.

To apply an illumination correction on the retinal image of healthy HRF database, we do not use a specific model for the luminance, because there is no specific need of precision. We therefore apply a naive approach, estimating the luminance by filtering the V channel of the image in the HSV colour space with a gaussian. We choose only the V channel because the largest amount of luminance information is carried by this channel. The σ of the gaussian has to be high to obtain a model of the luminance. The Illumination correction step is applied following the steps:

- i. convert the image I from RGB to HSV colourspace
- ii. obtain a model I_{Mv} of the illumination by filtering the H channel of the image with a gaussian
- iii. subtract the model to the the V channel of the image and add the value V_m . $\bar{I}_v = I_v - I_{Mv} + V_m$
- iv. convert the image \bar{I} from HSV to RGB color space.

The value V_m is the same for every image of the database, to make uniform the illumination between the various images. The value V_m is calculated as

the average between the values V_{mi} , where V_{mi} are the average values of the V channel in every image I_i .

In 3.16 there is an example of comparison between an original image with non uniform illumination, and the resulting image after the illumination correction step. Some artifacts are visible in the edges of the FOV, so this area is not considered for collecting the tiles.

3.4.3 Tiles collection

The collection of Tiles is used for the creation of a Dictionary, necessary for the Quilting algorithm. Since we want to create a linear vessel, when collecting the tiles we need to rotate them so that they share the same direction. Furthermore we have to organize the dictionary, grouping the tiles on the base of the calibre and the type of the vessel they are collected from. As we mentioned before we have created many types of different images maps and masks that contain useful informations for the tiles collections. To simplify the explanation of the algorithm used for the collection, we introduce the following symbols:

- Original image with luminance correction : IC ;
- arteries or veins skeleton map : $Skel$;
- Crossings map : $Cross$;
- Bifurcations map : Bif ;
- Calibre map: C ;
- Angle map: A ;

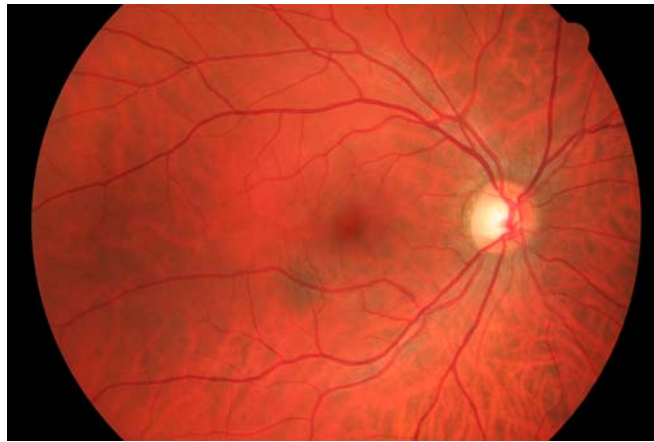
We will only show the collection procedure performed for one type of vessel, e.g. artery, since it is the same for the veins. First we mask the skeleton $Skel$ of the artery binary image with the bifurcations Bif and crossings $Cross$ maps. This step is needed to avoid collection in points that could possibly create artifacts along the linear vessel.

The skeleton image $\cap Skel$ is then computed for every pixel (x_p, y_p) set to white in the image resulting from the previous step. To avoid overlapping between collected tiles, another mask is computed where the pixels already collected are set to "white" and the other to "black". We call this mask CM .

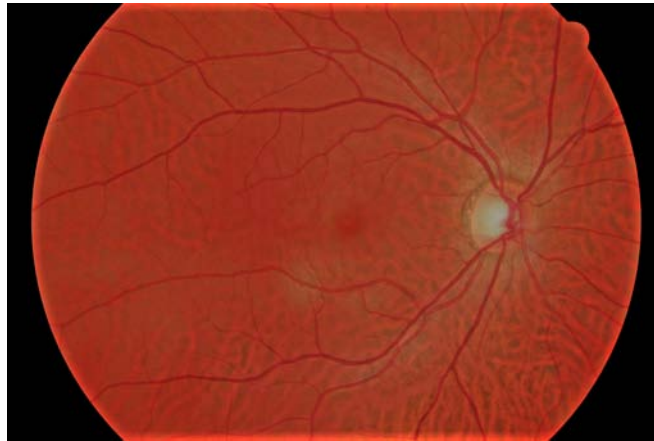
If the pixel (x_p, y_p) is black on the CM , then the calibre c and the angle α on that point are computed by looking at the corresponding pixel in A and C :

$$c = C(x_p, y_p)$$

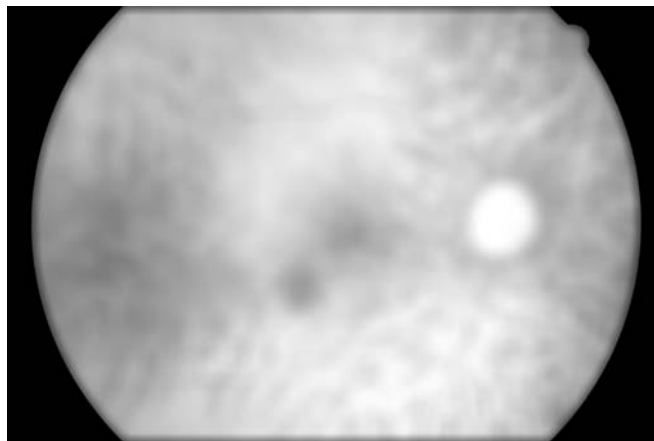
$$\alpha = A(x_p, y_p)$$



(a) Original Image



(b) Result of illumination correction



(c) Estimated Illumination

Figure 3.16: Illumination correction on a image of HRF database with the proposed algorithm

In relation to the angle α and the calibre c of the vessel in the pixel (x_p, y_p) a square neighbourhood N_p of the pixel p is computed on the $\cap Skel$ image. If in N_p more than one connected component is present, no tiles are collected from the point (x_p, y_p) . This constraint is added to avoid collection in vessels too close to each other, even if there is no bifurcation, due to the fact that the segmentation taken as ground truth is not perfect.

Another condition to collect a tile from N_p must be that the vessel has not to be too curved in that area. We therefore compute an index of curvature in N_p . considering that N_p is made of a small amount of pixels we considered that the angulature doesn't have to change a lot along the centerline in N_p :

$$v = |max(\alpha_{N_p}) - min(\alpha_{N_p})|$$

where v is the curvature index used and α_{N_p} is the collection of angles along the centerline in N_p .

The last constraint to add a tile in N_p is that we do not have to be too close to a branch edge point. In order not to be on an edge point, it's enough to have at least two white pixels in the contour of N_p . If the conditions mentioned above are all satisfied, then we can collect a tile from N_p . First we collect the corresponding pixels of N_p from the IC image, calling them N_{pIC} . Then N_{pIC} is rotated of angle α . considering the informations on the calibre, the tile is then collected from N_{pIC} , marking it with the vessel type from which it was picked and the calibre in point (x_p, y_p) . The last step after the tile collection is to update the CM mask. Only the pixels collected with the tile are marked as white in the CM mask, and not all the pixels of N_p .

Each tile has dimensions [n x m] where m is the same for every tile and is set to 10, to obtain a compromise between accuracy and computation time in quilting algorithm. The m dimension is parallel to the direction of the vessel. The dimension n, that is the one perpendicular to the direction of the vessel, differs from the calibre, due to the fact that we do not only collect tiles from the entire vessel width, but we slightly exceed into the background. This happens because the presence of the background will later help us to place the newly created vessel inside the image.

If w_{bg} is the half width of the background piece we want to collect on both side of the vessel and c is the calibre in point (x_p, y_p) then the tile will have n dimension equal to: $n = 2 w_{bg} c$.

3.4.4 Linear vessel generation

Once the tiles have been collected, we need to create a linear vessel. By linear we mean a vessel with a straight, non-curved centerline.

Building a linear vessel is the first step for the creation of any branch of the vessel trees that define the synthetic vasculature on a phantom retinal image. Afterwards, such linear vessel will be twisted to follow the actual

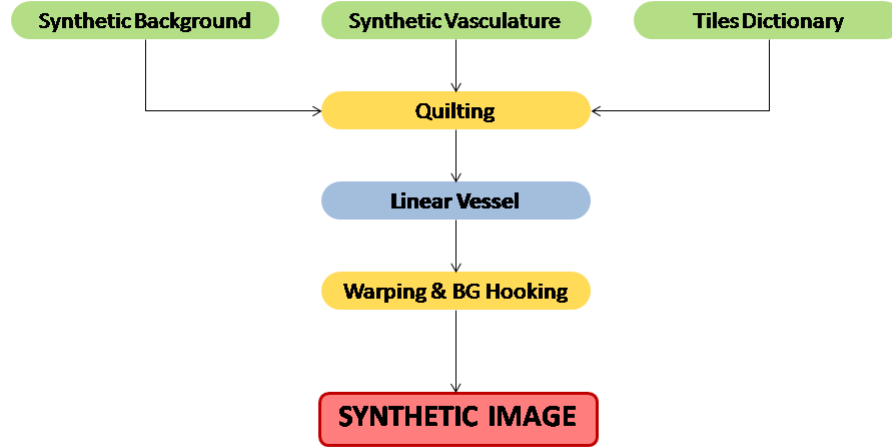


Figure 3.17: Synthetic retinal image algorithm block diagram

centerline of the considered branch. This happens for every branch of each one of the eight trees that define the vasculature. We compute the length of the centerline and we add the texture with the patch based algorithm in the direction of the vessel using tiles taken from the same vessel type of the branch and of the same calibre. The patch based algorithm works with the following steps:

- i. define a pool of tile of the right vessel type and calibre
- ii. pick up a random tile
- iii. if it is first, then place it
- iv. if not, then compute some compatibility coefficients with the previous tile (see [7] for more details) and, in a similar way, with the background.
- v. if those coefficients respect some constraints then place the tile and pass to the next one
- vi. if not, discard the tile and repeat from step ii.
- vii. if the number of iterations is more than a predetermined limit, then the algorithm goes in saturation; place the best tile found and pass to the next.

The placement of a tile is done roughly overlapping the two tiles and computing the best cut in the overlap area, made by the quilting technique. When a bifurcation occurs (thus when a branch has a parent) to build the

linear vessel the first tile is compared to the last tile of the parent cropped to the size of the son branch. It is like we put a phantom tile before the linear vessel, that is used only for the first comparison, but then is not placed. It has not be done more to manage the bifurcation areas, leaving this challenge for future implementations.

Because of the limited number of tiles, the algorithm goes very often in saturation. Thus the problem of artifacts in the vessel made that the illumination correction has made less visible. With the data currently available, it is not possible to reduce the occurrences of saturation phenomena, since this would require a higher number of tiles in the dictionary, meaning more images in the dataset. To avoid saturation, the challenge is to increase the number of tiles in the dictionary. Collect more tiles is not the only possible solution. Other solutions could be of using tiles of every calibre and rescaling them (but in total we have collected about 33000 tiles of arteries and of veins, again far away less than the 300000 tiles that is shown to be a good amount for the creation of a realistic background with the quilting technique). A third possible solution is to generate tiles using a model, increasing almost at infinity the number of tiles available, but losing the propriety of generating new vasculature from real data. For time reasons, none of these solutions have been implemented and could therefore be part of future work.

3.4.5 Warping and background hooking

Every time we create a linear vessel, this has to be warped and then placed on the background. To warp the linear vessel, first the angles along the centerline are computed. We proceed iteratively on every pixel profile of the linear vessel. A pixel profile can be defined as a matrix of pixels of dimension $[n \times 1]$ where n is the vessel calibre plus the background pixel width. Before adding the profile on the background, it has to be modified. Referring to the illumination correction done on the real images, suppose that no information about the difference in illumination within the vessel is carried, because all of the tiles has been collected from images with the same illumination value. The synthetic background instead has differences in the illumination. We can therefore use the information about illumination of the background to add again differences along the vessels, and in the same time make the illumination of the background consistent with the illumination of the vessel.

The corresponding profile of background pixels, on which the vessel profile has to be placed, is collected. Both profiles are converted from RGB colour space to HSV color space again. The V channel of the background pixels in the vessel profile and the corresponding pixels in the background profile are mediated. If we name:

- the pixel profile I ;
- the background pixels on I : I_k ;

- the background profile B ;
- the correspondent of I_k in B : B_k .

then:

$$I_{\{V\}} = I_{\{V\}} - (\bar{I}_{k_{\{V\}}} - \bar{B}_{k_{\{V\}}})$$

where \bar{X} refers to the average number and $X_{\{V\}}$ to the V channel.

Afterwards that the vessel profile is transformed again from HSV colour space to RGB colour space and placed on the background by substituting the background pixels with the vessel profile.

When all the vessels are completely placed on the background two more steps are needed to complete the image. First the vasculature is masked and the green and blue channel of the pixels are multiplied by two coefficients. The coefficient for the green channel is 0.5 and the one for the blue is 0.8. The coefficients are derived from empirical observations. Due to the lack of tiles and for the reverse illumination correction sometimes it can sometimes happen that some vessels are brighter than the background, and for this reason decreasing the green channel usually contributes to solve that problem. The blue channel is also decreased because, due to the conversion from RGB to HSV channel sometimes the vessels present a too high blue value. The last step is a smoothing along the edges of the vessel, because the simple replacement of the pixel profiles makes the transitions from background to vessel too sharp.

Chapter 4

Experiments and Results

In this chapter we will present the experiments that have been done on synthetic images and the relative results. The first evaluation is a visual comparison between real and synthetic retinal images. Since our final goal is to use the synthetic images to validate algorithms in retinal image analysis, we then checked if the behaviour of the synthetic images is comparable to the behaviour of real images when a typical analysis is performed. The first evaluation is a visual comparison between real and synthetic retinal images. Since our final goal is to use the synthetic images to validate algorithms in retinal image analysis, we then checked if the behaviour of the synthetic images is comparable to the behaviour of real images when a typical analysis is performed.

4.1 Visual Appearance

Our first goal was to create visually convincing retinal image phantoms. During the development of the present work every step done was conducted in order to achieve a realistic aspect. A visual inspection is an important first step to check the results of the algorithm, both because it gives hints on how to improve the algorithm itself and by allowing to conduct quantitative experiments on synthetic images.

A first comparison is performed between vessel trees.

Figures 4.1 and 4.2 show an example of comparison between four real vessel trees and four synthetic vessel trees. The first two are arteries and the last two are veins. The real trees have been taken from the arteries and veins binary masks, as explained in paragraph 3.3. The two images are close to each other: the branches have similar tortuosity and shape and the overall aspect of the tree is realistic, even if there is place for improvements. However from this first comparison it has to be said that there is a lot of room for future improvements. A first imprecision is the model chosen for

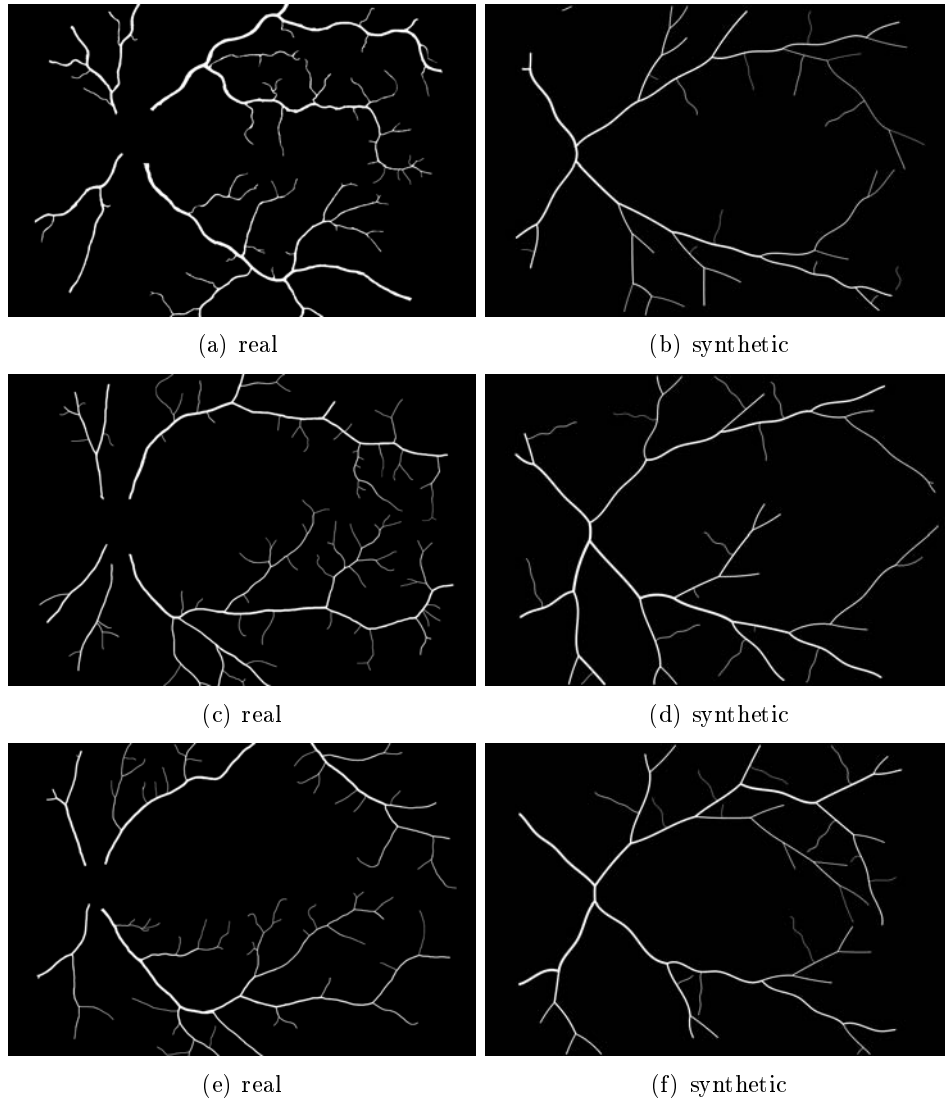


Figure 4.1: Comparisons between real and synthetic artery trees

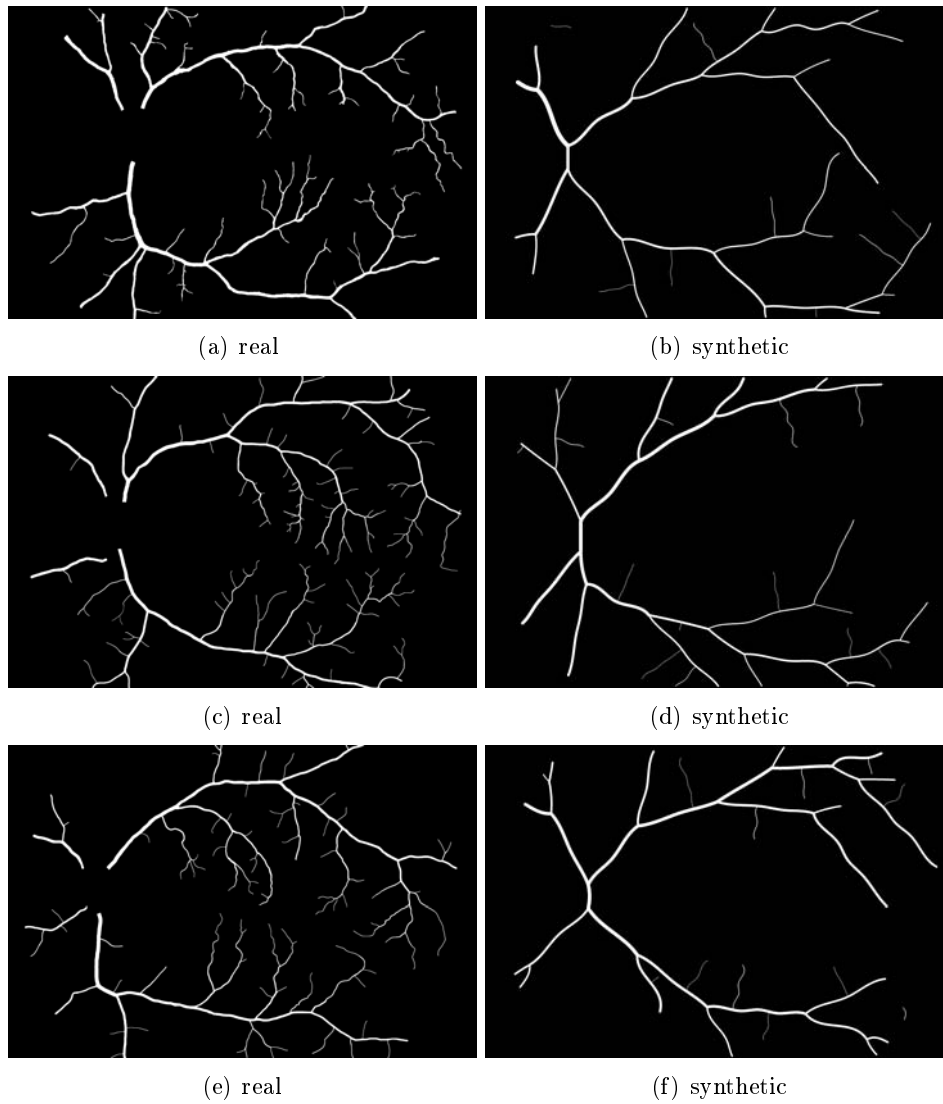


Figure 4.2: Comparisons between real and synthetic vein trees

the main shape of the branches. We use a parabolic MTA model, which is largely used in literature and fit for our purposes, but is not accurate to describe such vessel structures and loses validity when leaving the fovea region. To improve the aspect of the tree, in this sense a more accurate MTA model have to be developed.

From the comparison of real and synthetic images, two other inconsistencies appear:

- Firstly, the transition between parent and son branches may sometimes be abrupt and not physiological.
- Secondly the vessel calibre is sometimes too big around the fovea. This can happen because the calibre of the synthetic branches is computed using a model iteratively from the first branch, but in real images some bifurcation can be hidden along the vessels and the vessel calibre sometimes is reduced by physiological condition.

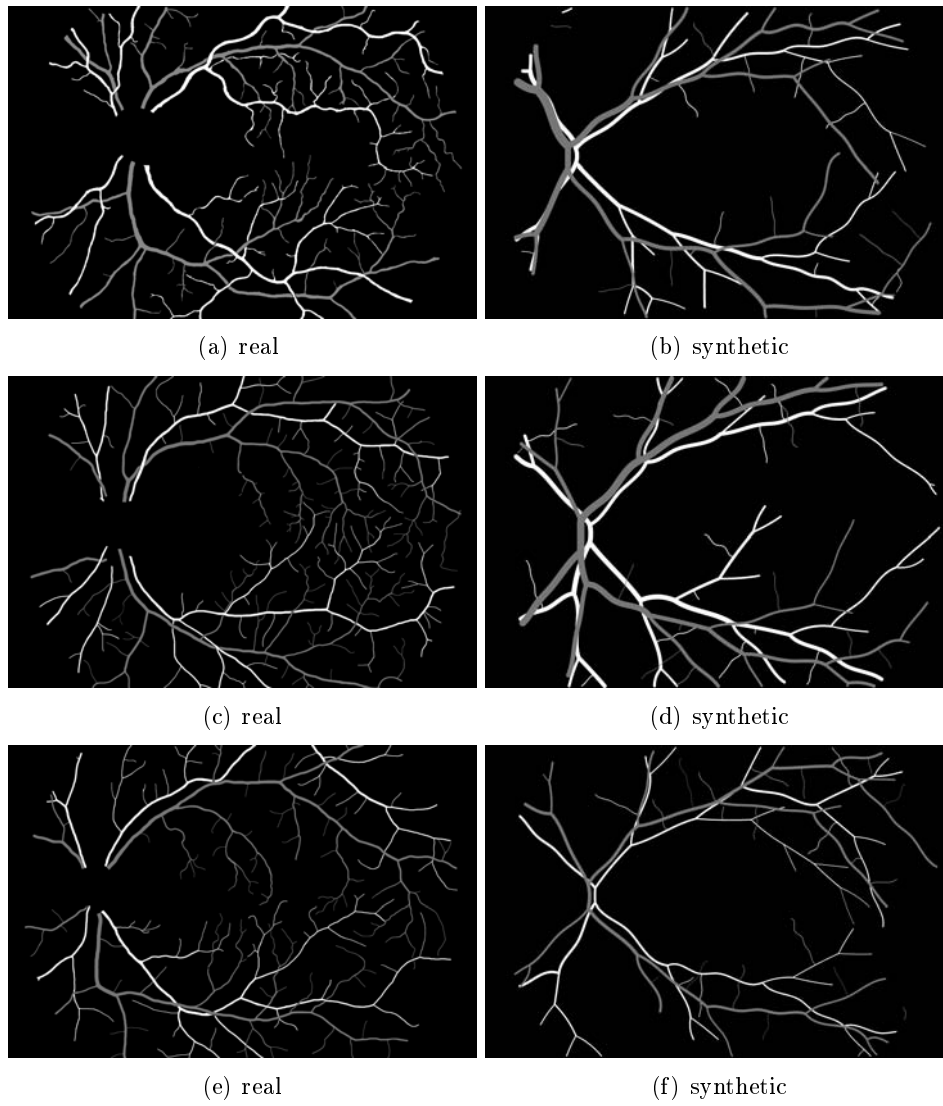


Figure 4.3: Comparisons between real and synthetic vasculature geometry

Let us now move on to consider a bigger scale. If we look at the entire vasculature system (figure 4.3), we can observe that the overall structure of the synthetic vasculature approaches the real one, but the smaller vessels are missing. This could partly be solved by improving model, vessel bifurcation and calibre, but other refinements can also be considered:

- including a higher number of trees. The retina was divided into four quadrants and therefore four trees because such is the number of major vessel arcades. Since the earliest bifurcations often happen in the OD area, however, the actual number of trees leaving the OD is higher.
- take in consideration the covering rate of the retina by the branches. Indeed there's not a total covering of the retina by the branches. This could be caused by the few number of tree coming out from the OD, but also could be because the directions of growth on the tree don't take in consideration the uncovered area. An improvement could be done for both of this aspects.
- including interactions between veins and arteries. We have constructed our model assuming that veins and arteries form two separate structures that are simply superposed in the image. In a physical environment, however, such structures interact with each other, e.g. resulting in a slight direction change of the veins when they intersect and artery, supposedly because of the difference in blood pressure in the vessels.

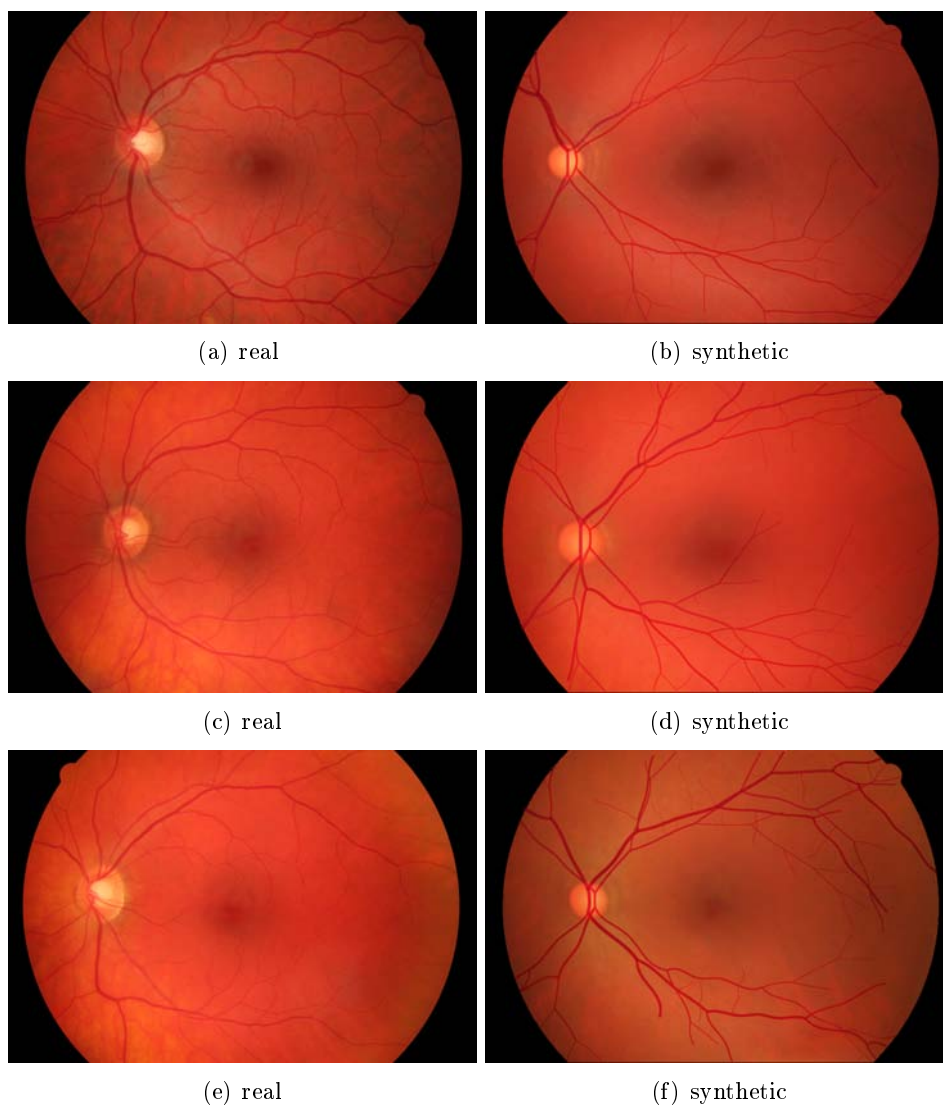


Figure 4.4: Comparisons between real and synthetic images

The last and most important comparison takes into consideration the entire vasculature system together with the background. The background is modelled with the algorithm detailed in [7]. The vasculature system is not just laid on the background, the two models are integrated to produce the result shown in Figure 4.4. The aspect of the complete synthetic images is close to the real images, including the central reflex present in some cases on vessels, which is an acquisition artifact and is generated in the synthetic images when generating the texture. Moreover, the illumination of the vasculature follows that of the background, brighter in the center, around the OD, and darker in proximity of the contour of the FOV. This shows that the two models can be properly combined to obtain images that is close to real ones, which was the purpose of the parallel projects.

Even if all the consideration made for the vasculature geometry should highly improve the overall aspect of the vasculature, there is still a lot of room for improvements. The bifurcation points, sometimes are not homogeneous with the rest of the vessel, due to the fact that our algorithm is not suited for the bifurcation points. Also the color of the vessel sometime is not homogeneous among the vasculature. This should be first a consequence of the lack of tiles: when the algorithm does not found a compatible tile, it places the best. This causes artifacts and repentine changes of colour in the vessels, that has been reduced with the Illumination Correction step, but not completely eliminated. A third aspect that should be improved is the hooking with the background, especially the smoothing step.

4.2 Vessel Detection

As we have widely said the final goal of our work was to generate ground truth for algorithms validation. The first goal to obtain is that at least our synthetic images behave in the same way of real images when processed with algorithms for retinal image analysis. The following experiment was conducted to prove that the vasculature segmentation on a set of synthetic images is correctly segmented with a segmentation algorithm. For the segmentation we have chosen the Soares [35], in the version that is currently available in VAMPIRE software. We have chosen this algorithm because it has been proven to be a stable and valid method for segmentation in retinal images. This test consist in a performance comparison of vessel detector on two different dataset:

- 15 real images from healty HRF dataset;
- 15 sinthetic images

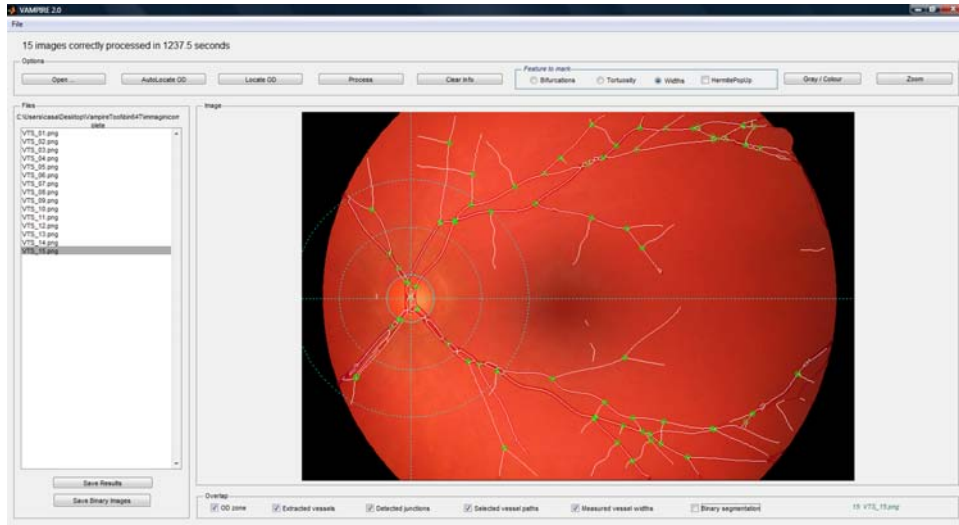


Figure 4.5: Example of VAMPIRE working on a synthetic image

The performances are compared using the following parameters:

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

$$\text{True Positive Rate} = TPR = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

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

Average and standard deviation for the values above for both real and synthetic images and the absolute difference of the average values between the two dataset can be found in table 4.1

Looking at table 4.1 we notice that the segmentation of synthetic phantoms outperforms a few the segmentation of HRF images. The FPR is lower and the Acc is higher. This let us think that synthetic vessel are segmented better than the real vasculature. This could be due to the method for synthetic vasculature and background hooking. This result allow us to identify where to continue the work in the future.

Table 4.1: Performance comparison

Parameters	HRF		Synthetic		
	average	std.	average	std.	a.d.
FPR	0.0214	0.0092	0.0138	0.0102	0.0076
TPR	0.7121	0.0415	0.7062	0.1347	0.0059
Acc	0.9538	0.0107	0.9665	0.0106	0.0127
Sp	0.9786	0.0092	0.9862	0.0102	0.0076
PPN	0.9706	0.0057	0.9784	0.0155	0.0078
PPV	0.7777	0.0621	0.7763	0.1450	0.0014

This experiment has the aim to show that the synthetic retinal images resulting from our method are not misleading by the existent segmentation algorithm, giving deeper quantitative experiments to future work, especially when the appearance of the synthetic vasculature will be more realistic.

Chapter 5

Conclusion and Future Work

In this thesis a novel method for synthesizing retinal phantoms is presented, the aim of which is to generate images with characteristics similar to those of a given dataset. The work is focused on the vasculature region and on its relationship with the background. A model-based approach is used for the vascular network geometry. Many features of the vessel tree, like major temporal arcade, bifurcations, first vessel calibre, length and tortuosity, are modeled. The parameters are estimated using statistical analysis on real images as well as visual inspection and then calibrated for arteries and veins. A control step has been added in order to avoid crossings between arteries or between veins and to prevent vessels from growing in the macula region. The texture is added on the vasculature network using a patch-based approach. The tiles are collected from real vascular regions, orienting them in the local vessel direction and are organized according to type and calibre. Through all these steps, the vessel structure is gradually built with anatomy and physiology related constraints, from the big arteries and veins that leave the OD region until the small vessels in the capillary bed. Such structure is then combined with the synthetic background in order to create an entirely synthetic image of the vascular system of the retina and its environment.

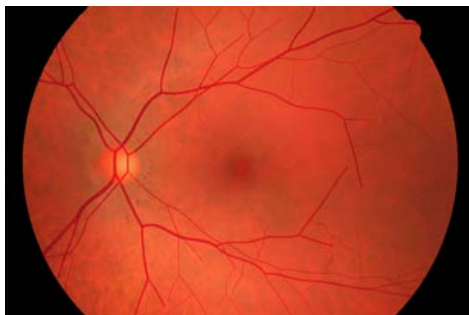


Figure 5.1: Example of Retinal Fundus Image Phantom

We can conclude that the model we constructed to simulate the vascular system of the retina is able to reproduce the main vessels and the first branches, locating bifurcations and crossings; it combines well with the background model, reproducing the illumination conditions and even the artifact of central reflex present in the real images. Since it was a preliminary work, a lot of improvements can clearly be made. As we have previously extendedly explained, the arcades should be described with a more appropriate model; bifurcations should be made smoother; the number of trees should be higher; physiological conditions and physical interactions should be taken into account. Moreover, we would need a larger dataset in order to be able to reduce the artifacts due to discontinuities in tiles overlay in the quilting algorithm. All of this could be part of future work on this topic, that would lead to a realistic model able to correctly reproduce the vascular system of the retina that might possibly be used, in the future, as in vitro test subject.

Bibliography

- [1] M.D. Abramoff, M.K. Garvin, and M. Sonka. Retinal imaging and image analysis. *Biomedical Engineering, IEEE Reviews in*, 3:169–208, 2010.
- [2] John A. Adam. Blood vessel branching: Beyond the standard calculus problem. *Mathematics Magazine*, 84:196–207, 2012.
- [3] B. Al-Diri and A. Hunter. Automated measurements of retinal bifurcations. *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, 25/11:205–208, 2009.
- [4] B. Al-Diri, A. Hunter, D. Steel, and Maged Habib. Manual measurement of retinal bifurcation features. *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4760 – 4764, 2010.
- [5] A. Budai, J. Hornegger, and G. Michelson. Multiscale approach for blood vessel segmentation on retinal fundus images. *Invest. Ophthalmol. Vis. Sci.*, 50(5):325, 2009.
- [6] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 341–346, New York, NY, USA, 2001. ACM.
- [7] S. Fiorini. Automatic generation of retinal fundus image phantoms: non - vascular regions. *Master Thesis, University of Padova*, 2014.
- [8] A. D. Fleming, S. Philip, K. A. Goatman, J. A. Olson, and P. F. Sharp. Automated assessment of diabetic retinal image quality based on clarity and field definition. *Investigative Ophthalmology and Visual Science*, 47(3):1120–1125, 2006.
- [9] M. Foracchia, E. Grisan, and A. Ruggeri. Detection of optic disc in retinal images by means of a geometrical model of vessel structure. *Medical Imaging, IEEE Transactions on*, 23(10):1189–1195, Oct 2004.

- [10] H. Freeman. On the encoding of arbitrary geometric configurations. *Electronic Computers, IRE Transactions on*, EC-10(2):260–268, June 1961.
- [11] V.V. Gafiychuk and I.A. Lubashevsky. On the principles of the vascular network branching. *Journal of Theoretical Biology*, 212(1):1 – 9, 2001.
- [12] 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*, pages 2145–2148, Sept 2011.
- [13] E. Grisan, A. Giani, E. Ceseracciu, and A. Ruggeri. Model-based illumination correction in retinal images. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 984–987, April 2006.
- [14] Z. Kókai, G. and Tóth and R. Ványi. Modelling blood vessels of the eye with parametric l-systems using evolutionary algorithms. In *AIMDM'99*, pages 433–442, 1999.
- [15] B. Kochner, D. Schuhmann, M. Michaelis, G. Mann, and K.-H. Engelmeier. Course tracking and contour extraction of retinal vessels from color fundus photographs: most efficient use of steerable filters for model-based image analysis. In K. M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 755–761, June 1998.
- [16] M. Kociński, A. Klepaczko, A. Materka, M Chekenya, and A. Lunder-vold. 3d image texture analysis of simulated and real-world vascular trees. *Computer Methods and Programs in Biomedicine*, 107(2):140 – 154, 2012.
- [17] M. Kretowski, Y. Rolland, J. B.-W., and J.-L. Coatrieux. Fast algorithm for 3-d vascular tree modeling. *Computer methods and programs in biomedicine*, 70(2):129–136, 2003.
- [18] Louisa Lam, S.-W. Lee, and C.Y. Suen. Thinning methodologies-a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(9):869–885, Sep 1992.
- [19] A. Lisowska, R. Annunziata, E. Trucco, D. Karl, and G.K. Loh. An experimental assessment of five indices of retinal vessel tortuosity with the ret-tort public dataset. *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'14)* , Chicago, US, August 2014 (in press).

- [20] X. Liu, H. Liu, A. Hao, and Q. Zhao. Simulation of blood vessels for surgery simulators. In *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, pages 377–380, April 2010.
- [21] C A. Lupaşcu, D. Tegolo, and E. Trucco. Accurate estimation of retinal vessel width using bagged decision trees and an extended multiresolution hermite model. *Medical Image Analysis*, 17(8):1164 – 1180, 2013.
- [22] M. Mescam, M. Kretowski, and J. Bezy-Wendling. Multiscale model of liver dce-mri towards a better understanding of tumor complexity. *Medical Imaging, IEEE Transactions on*, 29(3):699–707, March 2010.
- [23] Cecil D. Murray. The physiological principle of minimum work applied to the angle of branching of arteries. *J Gen Physiol*, 9(6):835–841, 1926.
- [24] Jan Odstřicilik. Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database. *IET Image Processing*, 7:373–383(10), June 2013.
- [25] F. Oloumi, R. M. Rangayyan, and A. L. Ells. Parabolic modeling of the major temporal arcade in retinal fundus images. *IEEE T. Instrumentation and Measurement*, 61(7):1825–1838, 2012.
- [26] F. Oloumi, R.M. Rangayyan, and A.L. Ells. Parametric representation of the retinal temporal arcade. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1–4, Nov 2010.
- [27] F. Oloumi, R.M. Rangayyan, and A.L. Ells. Dual-parabolic modeling of the superior and the inferior temporal arcades in fundus images of the retina. In *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, pages 1–6, May 2011.
- [28] M. S. Olufsen, C. S. Peskin, W. Kim, E. M. Pedersen, A. Nadim, and J. Larsen. Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions. *Annals of Biomedical Engineering*, 28(11):1281–1299, 2000.
- [29] Niall Patton, Tariq M. Aslam, Thomas MacGillivray, Ian J. Deary, Baljean Dhillon, Robert H. Eikelboom, Kanagasingam Yogesan, and Ian J. Constable. Retinal image analysis: Concepts, applications and potential. *Progress in Retinal and Eye Research*, 25(1):99 – 127, 2006.
- [30] A. Perez-Rovira, T. MacGillivray, E. Trucco, K.S. Chin, K. Zutis, C. Lupaşcu, D. Tegolo, A. Giachetti, P.J. Wilson, A. Doney, and B. Dhillon. 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*, pages 3391–3394, Aug 2011.
- [31] D. Relan, T. MacGillivray, L. Ballerini, and E. Trucco. Retinal vessel classification: Sorting arteries and veins. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 7396–7399, July 2013.
- [32] W. Roux. doctoral thesis on the bifurcation of blood vessels, cited by thompson [37].
- [33] 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, SIGGRAPH '94*, pages 205–212, New York, NY, USA, 1994. ACM.
- [34] W. Schreiner and P.F. Buxbaum. Computer-optimization of vascular trees. *Biomedical Engineering, IEEE Transactions on*, 40(5):482–491, May 1993.
- [35] J.V.B. Soares, J.J.G. Leandro, R.M. Cesar, H.F. Jelinek, and M.J. Cree. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. *Medical Imaging, IEEE Transactions on*, 25(9):1214–1222, Sept 2006.
- [36] B. N. Steele, M. S. Olufsen, and C. A. Taylor. Fractal network model for simulating abdominal and lower extremity blood flow during resting and exercise conditions. *Computer Methods in Biomechanics and Biomedical Engineering*, 10(1):39–51, 2007. PMID: 18651270.
- [37] D’Arcy W. Thompson. *On Growth and Form*. Dover Publications, June 1992.
- [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*, pages 7152–7155, July 2013.
- [39] E. Trucco, A. Ruggeri, T. Karnowski, L. Giancardo, E. Chaum, J. P. Hubschman, B. al Diri, C. Y. Cheung, D. Wong, M. Abramoff, G. Lim, D. Kumar, P. Burlina, N. M. Bressler, H. F. Jelinek, F. Meriaudeau, G. Quellec, T. MacGillivray, and B. Dhillon. Validating retinal fundus image analysis algorithms: Issues and a proposal. *Investigative Ophthalmology and Visual Science*, 54(5):3546–3559, 2013.

- [40] N. Tsafnat, G. Tsafnat, and T.D. Lambert. A three-dimensional fractal model of tumour vasculature. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 1, pages 683–686, Sept 2004.
- [41] Yangfan Wang, Guangrong Ji, Ping Lin, and Emanuele Trucco. Retinal vessel segmentation using multiwavelet kernels and multiscale hierarchical decomposition. *Pattern Recognition*, 46(8):2117 – 2133, 2013.
- [42] M. Zamir, J. A. Medeiros, and T. K. Cunningham. Arterial bifurcations in the human retina. *J Gen Physiol*, 74:537–548, 1979.

Acknowledgements

My sincere gratitude goes to my supervisor Prof. Manuel Trucco, for his support and guidance during my experience in the CVIP group of the University of Dundee; many thanks for all the advices (and encouraging words) you gave me.

I would also like to express my deepest thanks to all the VAMPIRE team: Manuel, Lucia, Kris, Enrico and Roberto, without whom this dissertation would not have been possible. A special thank goes to Lucia, who, even from distance, has helped me more than the necessary.

Maddi