

POLITECNICO DI TORINO



Master Degree course in Biomedical engineering

Master Degree Thesis

A skeletonization approach for the evaluation of vascular complexity using in-vitro phantoms and 3D LED-based photoacoustic images

Supervisors

prof. Filippo Molinari

prof. Kristen M. Meiburger

Candidate

Roberta BRUSCHETTA

ACADEMIC YEAR 2018 – 2019

Abstract

Vasculature analysis is a fundamental aspect in diagnosis, treatment, outcomes evaluation and follow-up in several medical fields.

Since vessels branching, maturation or quiescence depend on microenvironment and angiogenic signals, vascular attributes are deeply affected by most of the diseases, from a simple cold to the tumours. Moreover, it was demonstrated that efficient therapies administration leads to attributes normalization.

Therefore, vascular network characterization can be a powerful means for earlier pathologies revealing and for their monitoring. For this reason, the development of a *non-invasive* and *quantitative* method for the evaluation of blood vessels complexity is a very important issue.

Many imaging techniques can be used for visualizing blood vessels, but their application is limited by problems such as high costs, need of contrast agents and use of ionizing radiations.

Photoacoustic imaging (PAI) is an emerging hybrid modality, which makes use of optical excitation and ultrasound detection. This non-invasive and non-ionizing technique combines the qualities of both the methods: the contrast and the spectral specificity of optical imaging and the high penetration depth and the spatial resolution of acoustic imaging. The optical and radio-frequency waves used in this technique, enable the possibility to use endogenous agents such as haemoglobin thanks to its absorption properties. Moreover, the most recent LED-based systems are economic, safe and portable.

Starting from biomedical images, many techniques have been developed to extract and represent vessels structure. They differ among them for the pre-processing steps, the computational time, the accuracy and the visual quality in results showing.

Skeletonization is a very efficient method to evaluate vasculature architecture and has been used with 3D contrast enhanced ultrasound images to characterize thyroid nodular vessels and tumour vasculature in vivo studies.

The aim of this thesis is the design and manufacturing of in-vitro vessel phantoms to evaluate the feasibility of using 3D LED-based photoacoustic images for vascular complexity quantification using a skeletonization algorithm.

In particular, three *in-vitro* agar phantoms with different vessels complexity were designed using the CAD Software *Solidworks* and realised with the use of a 3D printer.

3D images acquisition was performed with a LED-based photoacoustic imaging system of *Prerion Corporation*.

Then, the images obtained were segmented and the three-dimensional volumes were reconstructed. Automatic segmentation performances were evaluated through a comparison with a manual segmentation.

From the reconstructed volumes, a 3D skeleton, based on medial axis extraction, was calculated and vascular parameters were measured. In particular, three morphological and three tortuosity parameters were computed: number of vascular trees (NT), vascular density (VD), number of branches (NB), distance metric (DM), inflection count metric (ICM) and sum of angles metric

(SOAM). In addition, mean radius (MR) was calculated.

Outcomes were compared with the ground truth obtained converting the original models from *Solidworks* to MATLAB.

Analysing results, it's possible to conclude the validity of the skeletonization from 3D LED-based photoacoustic images as approach for vascular complexity quantification.

Table of contents

1	Introduction	1
2	Photoacoustic Imaging	3
2.1	Photoacoustic effect and imaging principles	3
2.1.1	Photoacoustic image contrast	4
2.1.2	PAI and Haemoglobin	5
2.1.3	Penetration depth	6
2.1.4	Spatial Resolution	7
2.1.5	Ultrasounds and imaging	7
2.1.6	Legislation	7
2.2	Prexion System	8
3	Blood Vessels Extraction Techniques	10
3.1	Pattern Recognition Techniques	10
3.1.1	Multi-scale approaches (MSA)	10
3.1.2	Skeleton-based approaches (SBA)	10
3.1.3	Ridge-based approaches (RBA)	12
3.1.4	Region Growing approaches	13
3.1.5	Differential Geometry-based Approaches (DG)	13
3.1.6	Matching Filters Approaches	14
3.1.7	Global Thresholding	15
3.1.8	Mathematical Morphology Schemes	16
3.2	Model-based Techniques	17
3.2.1	Deformable Models (DM)	17
3.2.2	Parametric Models (PM)	19
3.2.3	Template Matching	19
3.3	Machine Learning	19
4	Phantoms	22
4.1	Phantoms importance in biomedical imaging	22
4.2	Phantoms design and fabrication	23
4.2.1	Solidworks design	23
4.2.2	3D Printing	25
4.2.3	Phantoms Realisation	27

5	Data Acquisition and Processing	30
5.1	Images Acquisition	30
5.1.1	Setup for the acquisition	30
5.1.2	Prexion settings	30
5.2	Segmentation	32
5.2.1	Automatic Segmentation	32
5.2.2	Manual Segmentation	38
5.2.3	Validation	40
5.3	Models Conversion from STL to MATLAB	43
5.3.1	Standard Tessellation Language	43
5.3.2	Function VOXELISE	44
6	Skeletonization and Parameters Calculation	51
6.1	Skeletonization	51
6.1.1	Function bwskel	51
6.1.2	Skeleton cleaning	55
6.1.3	Branchpoints search	60
6.1.4	Endpoints search	62
6.2	Breadth First Search	66
6.3	Parameters	69
6.3.1	Tortuosity Parameters	69
6.3.2	Morphological Parameters	75
6.3.3	Mean Radius	75
7	Results	80
8	Interpolation	86
9	Conclusion and future developments	88
	Bibliography	89
	Acknowledgments	I

List of figures

2.1	Photoacoustic imaging operating principle: (A) tissue is irradiated from a laser pulse and local heating is produced by the absorbed energy. (B) thermoelastic expansion generates ultrasounds that are detected outside the sample. (C) Flowchart with steps. [1]	4
2.2	Absorption coefficient spectra of endogenous tissue chromophores. Oxyhaemoglobin (HbO ₂) in red, deoxyhaemoglobin (HHb) in blue, water in black, lipid ^(a) in brown, lipid ^(b) in pink, melanin in black dashes, collagen in green and elastin in yellow. [2]	5
2.3	Absorption coefficient spectra of Oxyhaemoglobin (HbO ₂) and deoxyhaemoglobin (HHb). [3]	6
2.4	(A) PAI System of Prexion Corporation, (B) PLED-PAI probe with imaging plane with LED array design in the insert, (C) Scheme of PAI system.[4][5]	9
3.1	Example of thinning procedure - background black pixels and foreground white pixels: (A) Original object (B) Finished skeleton.[6]	11
3.2	Example of Distance Transform: (A) Object indicated in coloured pixels (B) Distance Transform calculated with the mask on right (moving only on rows and columns).[7]	12
3.3	Example of vessels skeletonization: on the left the starting mask and on the right the final skeleton.[8]	12
3.4	Vessel tree extracted from mouse clicks.[9]	13
3.5	Region Growing procedure: (A) Start of growing region (B) Growing process after few iterations.[7]	13
3.6	(A) DSA image of the cerebral vessels (B) Vessel detection using four scales.[9] . .	14
3.7	Corrupted image with salt and pepper noise on the left, enhanced image with a 3 by 3 median filter on the right.[10]	15
3.8	Bi-modal luminance histogram with separation of background and object.[7]	15
3.9	Original volume on the left and filtered MR images on the right.[9]	16
3.10	Example of morphological operations using a disk with ray 5. [7]	17
3.11	Segmentation of thoracic aorta with a parametric deformable model.[7]	18
3.12	Segmentation of retinal vessel with a snake.[7]	18
3.13	Level set implementation: function in the higher dimensional space on the right and its K level on the left.[7]	19
3.14	(A) Scheme of Unsupervised learning approach (B) Scheme of Supervised learning approach.[11]	20
3.15	Model of a neuron.[12]	21
3.16	Convolutional Neural Network for vascular segmentation.[11]	21
4.1	Principal Solidworks functions used.	23

4.2	Models with the indication of external diameters - (A) M_{simple} (B) $M_{complex}$ (C) M_{deep} .	24
4.3	M_{simple} in Solidworks - (A) Front view (B) Lateral view (C) Section view.	24
4.4	$M_{complex}$ in Solidworks - (A) Front view (B) Lateral view (C) Section view.	24
4.5	M_{deep} in Solidworks - (A) Front view (B) Lateral view (C) Section view.	25
4.6	(A) Project MJP 2500 Plus (B) MJP EasyClean System (C) MJP EasyClean System open. [13]	25
4.7	Photograph of the process of printing.	26
4.8	Printed models before wax removal - (A) Top view (B) Front view.	26
4.9	Printed models - (A) $M_{complex}$ (B) M_{simple} (C) M_{deep} (D) Zoom on M_{simple} .	27
4.10	Ink test - (A) Phantoms with four different inks above (B) corresponding acquired image below.	28
4.11	Models filled with black ink.	28
4.12	Agar preparation steps - (a) Agar weighted with a digital balance (b) Powder put in a small pot with the water (c) Stirring (d) Solution is brought to the boil (e) Solution pouring in a container (f) Cooling down.[4][5]	29
4.13	Finished phantoms - (a) M_{simple} (b) $M_{complex}$ (c) M_{deep} .	29
5.1	(A) Setup for images acquisition (B) Zoom on probe and phantom.	30
5.2	Interface for the visualization and the acquisition of images, in yellow the buttons for the acquisition and in red the screen with the settings.	31
5.3	Principal steps of automatic segmentation - (A) Original image (B) Cropped grayscale image (C) Median filtered image (D) Closed image (E) Thresholded image (F) Final mask.	33
5.4	M_{simple} : (A) Original model (B) model after median filter (C) segmented model.	34
5.5	$M_{complex}$: (A) Original model (B) model after median filter (C) segmented model.	35
5.6	M_{deep} : (A) Original model (B) model after median filter (C) segmented model.	36
5.7	Automatic Segmentation Flowchart.	37
5.8	(A) Dialog box for the manual segmentation (B) Segmented image with red boundaries.	38
5.9	Comparison among automatic (A) and manual (B) segmentation.	38
5.10	Manual Segmentation Flowchart.	39
5.11	Graphic definition of VP, VN, FP and FN.	40
5.12	Segmentation performance evaluation for the three models: M_{simple} in green, $M_{complex}$ in blue, M_{deep} in yellow.	42
5.13	Example of an STL representation of a spherical solid object.[14]	43
5.14	(A) Facet normal definition from vertices (B) Example of an STL structure.[14]	43
5.15	(A) Vertices matrix (B) Normals matrix.[14]	44
5.16	Reorientation of the model from Solidworks to MATLAB.[14]	45
5.17	Scheme of voxellisation procedure and some possible errors.[15]	47
5.18	Some errors due to thin sections.[16]	47
5.19	Voxelise Flowchart.	48
5.20	MIP of converted M_{simple} from Solidworks.	49
5.21	MIP of converted $M_{complex}$ from Solidworks.	49
5.22	MIP of converted M_{deep} from Solidworks.	50
5.23	Converted models: (A) M_{simple} (B) $M_{complex}$ (C) M_{deep} .	50
6.1	26-neighbourhood for the point v : indices representation (a) and other two representations (b) and (c). [17]	51
6.2	Tree representation of the 26-neighbourhood $N(v)$. [17]	52

6.3	The six directions of the sub-cycles.[17]	53
6.4	Skeletonization flowchart.	54
6.5	Points agglomeration in the skeleton on the left and its elimination on the right.	55
6.6	M_{simple} - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.	56
6.7	$M_{complex}$ - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.	56
6.8	M_{deep} - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.	56
6.9	M_{simple} : (A) Original model and (B) corresponding skeleton.	57
6.10	$M_{complex}$: (A) Original model and (B) corresponding skeleton.	57
6.11	M_{deep} : (A) Original model and (B) corresponding skeleton.	57
6.12	M_{simple} : (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.	58
6.13	$M_{complex}$: (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.	58
6.14	M_{deep} : (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.	58
6.15	Skeleton Cleaning Flowchart.	59
6.16	(A) Example where the point with three neighbours is a branchpoint (B) Example where the point with three neighbours isn't a branchpoint.	60
6.17	(A) Example of a branchpoint with three branches, the branchpoint is the blue voxel, its neighbours are the yellow, while the red are the neighbours of the neighbours (B) Matrix with the neighbours to control if there are three separated connected objects.	60
6.18	Branchpoints search flowchart.	61
6.19	(A) Example where the endpoint is correctly individuated (B) Example where the endpoint is missed because it has two neighbours.	62
6.20	Matrix indices for the neighbourhood and the determination of the endpoints - (A) Before slice (B) Pixel slice (C) After slice.	62
6.21	Endpoints search flowchart.	63
6.22	M_{simple} - Branchpoints in red and Endpoints in green.	64
6.23	$M_{complex}$ - Branchpoints in red and Endpoints in green.	64
6.24	M_{deep} - Branchpoints in red and Endpoints in green.	65
6.25	Raster image (a) seen as a graph (b).	66
6.26	BFS algorithm on the M_{simple} - In order, the individuation of the six branches (red, green, blue, yellow, cyan and magenta).	67
6.27	Breadth First Search Flowchart.	68
6.28	Example of type I tortuosity in the basilar artery shown at left. At right, three normal subjects' vessels are reported. This abnormality produces cranial nerve deficits.[18]	70
6.29	Example of type II tortuosity in an AVM nidus. Vessels with different curvatures and lengths are packed in a small volume. Vessels inside the tumour are red, vessels that pass or supply it are yellow and healthy vessels outside it are blue. At left the nidus is shown with a full opacity, in the centre with a partial opacity and at right the vessels are alone.[18]	70
6.30	Example of type III tortuosity in a malignant glioma. Colours legend is the same of the previous figure. Vessels inside the tumour have high frequency and low amplitude coils or sinusoidal patterns.[18]	70

6.31	Distance metric defined as the sum of distances between adjacent 3D points along actual vessel path (indicated with short narrows) divided by the straight path between the endpoints (long narrow).[18]	71
6.32	Geometric implementation of the Frenet Frame.[18]	72
6.33	Recognition of inflection points in a 3D sine.[18]	73
6.34	In-plane angle Φ for point P_k . [18]	74
6.35	Schematic representation of the three tortuosity parameters - (A) DM (B) SOAM (C) ICM.[19]	75
6.36	Comparison among different distance metrics.[20]	76
6.37	(a) Skeleton of M_{simple} , (b) <i>bwdistgeodesic</i> of M_{simple} , (c) Slice 250 of <i>bwdistgeodesic</i> of M_{simple} (d) Slice 250 of perimeter <i>bwdistgeodesic</i> of M_{simple}	77
6.38	Radius value calculated in different points has to be converted differently.	77
6.39	Determination of resolution in acquired images.	78
7.1	Comparison of results among acquired models and ground truth.	84
7.2	Comparison among the different methods for mean radius calculation on the original model respect to the real value calculated manually in Solidworks.	85
7.3	Comparison among MR calculated with Euclidean method, on automatic models respect original ones.	85
8.1	M_{simple} - (A) Original skeleton from acquired model (B) Interpolated skeleton MIP (C) Cleaned skeleton MIP.	86
8.2	Comparison among original skeleton in green, automatic reconstructed skeleton in blue and interpolated skeleton in yellow.	87

List of tables

5.1	Validation segmentation M_{simple} .	41
5.2	Validation segmentation $M_{complex}$.	41
5.3	Validation segmentation M_{deep} .	41
5.4	Conversion Resolution M_{simple} .	44
5.5	Conversion Resolution $M_{complex}$.	45
5.6	Conversion Resolution M_{deep} .	45
6.1	Number of branchpoints and endpoints of the three models.	64
6.2	Abnormal Tortuosity Types.[18]	71
6.3	Conversion Factor Automatic Models.	78
6.4	Conversion Factor original models - M_{simple} on the left, $M_{complex}$ in the centre, M_{deep} on the right.	79
7.1	Results M_{simple} - Threshold 20.	80
7.2	Results M_{simple} - Threshold 50.	81
7.3	Results $M_{complex}$ - Threshold 20.	81
7.4	Results $M_{complex}$ - Threshold 50.	82
7.5	Results M_{deep} - Threshold 20.	82
7.6	Results M_{deep} - Threshold 50.	83
8.1	Comparison among original skeleton, automatic reconstructed skeleton and interpolated skeleton M_{simple} .	87

Chapter 1

Introduction

Thanks to blood vessels, oxygen and nutrients are provided to all the body and waste products are eliminated. Most of the diseases influence vessel attributes, such as their number, radius, tortuosity and branching pattern.[21]

For example, tumours induce the growth of vessels clusters with abnormal tortuosity, while chronic inflammations induce neo-angiogenesis. Vascular tortuosity increase and arteries narrowing are provoked by hypertension, diabetes and many other auto-immune pathologies. Even a simple cold modifies vasculature morphology with vasodilatation. Radius is altered by vasospasm and other vasculopathies, while angiogenesis is provoked by infections. Also stroke, aneurysm and arteriovenous malformations affect vessels morphology.[22][21] In addition, vessels shape normalizes if a successful therapy is administrated.[21]

Therefore, the possibility of a non-invasive and quantitative evaluation of 3D vessel attributes, could be a new powerful method of early diagnosis and staging diseases.[21]

Obviously, this type of analysis is based on some statistical evaluations about population vessel attributes in both healthy and pathological subjects.[21]

Many imaging techniques can be used for visualizing vasculature structure, but they have some disadvantages, such as high cost, need of contrast agents or use of ionizing radiations.[23]

For example, computed tomographic angiography (CTA) has an excellent spatial resolution and it is very widespread but it uses ionizing radiations and iodinated contrast agents, while magnetic resonance angiography (MRA) in spite of very good contrast and temporal resolutions and lack of ionizing radiation, suffers from rapid extravasation of the contrast agent that affects the accuracy. Doppler ultrasound imaging (DU) has good costs, large availability and it doesn't use nephrotoxic contrast agents, but it is operator-dependent, the few existing contrast agents have short duration and imaging problems occur for obese or with bowel gas patients. Also, a more recent technique, acoustic angiography, that uses dual-frequencies ultrasound transducers for the minimization of background,[24] needs exogenous agents and custom-made probes, while optical coherence tomography (OCT) has a limited penetration depth.[23][25]

Photoacoustic imaging is a imaging modality that goes a long way in the last decade, where ultrasounds are generated from the interaction between a pulsed light and the biological tissues irradiated. So, it is non-invasive and non-ionizing and it combines the high spatial resolution and the penetration depth of ultrasounds with the high contrast and the spectral specificity of optical imaging. In fact, ultrasound scattering in biological tissues is two or three order of magnitude weaker than the optical one.[26] [4]

In particular, LED-based Photoacoustic systems are inexpensive, compact, multi-wavelength and particularly stable.[4]

PAI exploits haemoglobin to observe blood vessels with a deep penetration without exogenous contrast agents. So, it can be used to monitor several biological tissues where microvascular changes are critical, such as breast, brain, skin and eyes.[23]

With photoacoustic imaging a lot of vessel structure data have been collected, but the studies about vessel architecture analysis aren't enough yet.[23]

Chapter 2

Photoacoustic Imaging

2.1 Photoacoustic effect and imaging principles

The *photoacoustic effect* was discovered by Alexander Graham Bell in 1880. He observed that absorption of modulated sunlight in media leads to sound generation. Then, for some years, research in this field proceeded very slowly until, in the 1960s, there was lasers development. This gave a great impulse to PA because of its interesting features such as high peak power, spectral purity, precision and directionality. In the following years, several industrial and scientific applications were proposed, but only in the 1990s the first biomedical images appeared. For very interesting results, it was necessary to wait the 2000s when an incessant development in terms of instrumentation, algorithms and applications started. Recently, lasers are being supplanted by LED, that have the same performance but are cheaper and safer.[2]

In *photoacoustic imaging*, modulated electromagnetic waveforms, often a nanosecond pulsed light, irradiate tissues of interest. The most commonly used wavelengths are that in the visible (400 nm - 700 nm) and the near infrared (700 nm - 1100 nm) because they are harmless for human tissues and they work well in terms of contrast and penetration.[27] The incident energy, depending on wavelength, penetrates to some depths and it's absorbed by tissues chromophores, such as haemoglobin, melanin, water and lipids and so it is converted to heat by vibrational and collisional relaxation. This local temperature increase, that must be small (less than 0.1 K) to not damage tissues, causes an increase of pressure for the *thermoelastic effect*, followed by a relaxation and in this way megahertz acoustic waves with low amplitude are emitted (less than 10 KPa). They propagate to the surface where are detected at different spatial points thanks to ultrasound receivers (Figure 2.1). Knowing the speed of sound, it's possible to reconstruct the image by measuring the time of ultrasounds arrival.[2]

There are several medical PAI application fields like, among others, oncology, cardiology, neurology, molecular imaging and ophthalmology.[4]

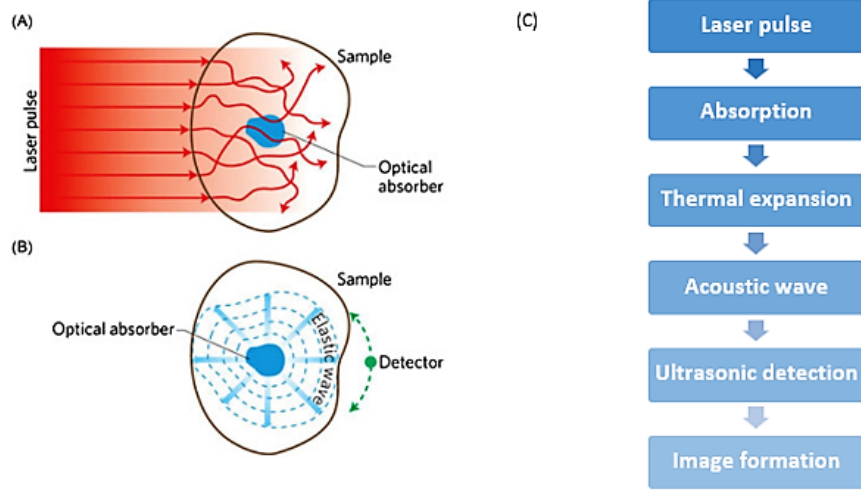


Figure 2.1: Photoacoustic imaging operating principle: (A) tissue is irradiated from a laser pulse and local heating is produced by the absorbed energy. (B) thermoelastic expansion generates ultrasounds that are detected outside the sample. (C) Flowchart with steps. [1]

2.1.1 Photoacoustic image contrast

A PA image represents the initial pressure distribution p_0 produced by the irradiation with the optical energy and it depends on optical absorption and scattering of tissues.[2]

With the assumption of the impulsive heating, therefore that acoustic propagation time is small compared to length scale of the heated volume, it's possible to make some thermodynamic considerations for which p_0 at a point r is proportional to the absorbed optical energy $H(r)$ with the law:

$$p_0(r) = \Gamma H(r) \quad (2.1)$$

Γ is the Grüneisen coefficient, a dimensionless thermodynamic constant that measures the conversion efficiency of heat to pressure and defined as $\Gamma = \beta \frac{c^2}{C_p}$ where β is the volume thermal expansivity, c the sound speed and C_p the specific heat capacity at constant pressure. The absorbed energy is the product of the local absorption coefficient $\mu_a(r)$ and the optical fluence $\varphi(s, \mu_a, \mu_s, g)$, where μ_a and μ_s are respectively absorption and scattering coefficients of the tissue and g is the anisotropy factor. However, in PA imaging mechanical and thermodynamic properties vary so weakly between different tissues to assume that they are spatially invariant. For this reason, even if Γ in a little part influences image contrast, it's assumed that it is due only to optical absorption and scattering.[2]

Differences in optical absorption between tissues are larger than acoustic impedance ones. In general, tuning laser excitation wavelength to the peak absorption of a specific tissue chromophore, the image contrast is selectively enhanced.[2]

It is possible to use endogenous or exogenous contrast agents. In fact, some endogenous molecules, like melanin and haemoglobin, have a strong absorption in the visible and near infrared regions,

providing an obvious source of primary contrast.[2]

In particular, both oxygenated and deoxygenated haemoglobin, show a preferential absorption from 650 to 900 nm, making PA particularly suited for microvasculature imaging. Melanin is highly localized in regions as skin or retina, it isn't the major constituent of tissues, so it is less used. Lipids absorption is lower for the wavelengths of interest and it becomes predominant at 1210 nm, so it is used to detect arteriosclerotic plaques.(Figure 2.2)[2]

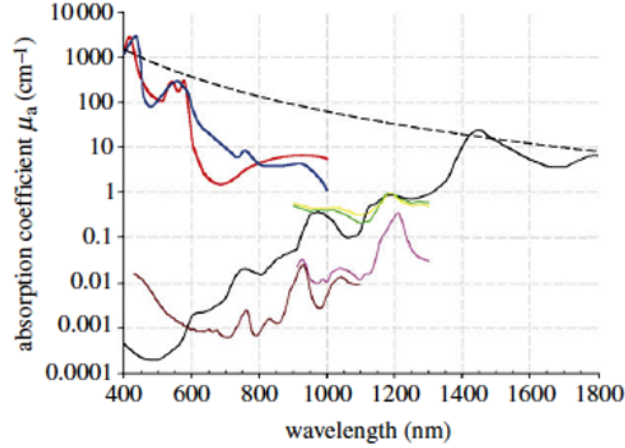


Figure 2.2: Absorption coefficient spectra of endogenous tissue chromophores. Oxyhaemoglobin (HbO_2) in red, deoxyhaemoglobin (HHb) in blue, water in black, lipid^(a) in brown, lipid^(b) in pink, melanin in black dashes, collagen in green and elastin in yellow. [2]

Endogenous agents are preferable because they are safe and they don't alter physiological parameters. However, in some applications, such as the detection of early stage tumours, exogenous agents are used in order to provide a biggest signal. Among the exogenous contrast agents more used, there are organic dyes like indocyanine green (ICG) and methylene blue, nanostructures as gold nanoparticles and carbon nanotubes, fluorescent proteins and quantum dots.[26] They can bind to specific cellular receptors providing the source of contrast for PA imaging.[2]

2.1.2 PAI and Haemoglobin

In haemoglobin absorption spectrum, there are changes depending on bindings. Oxygenated haemoglobin absorbs strongly until 600 nm and then drops off drastically, while deoxygenated haemoglobin decreases more slowly with increasing wavelengths.(Figure 2.3)[27]

Exploiting haemoglobin, with photoacoustic imaging, different vasculature parameters can be monitored such as structure, oxygenation status and flow. Therefore, this technique is suited for detection of tumours and pathologies with abnormalities in the morphology and function of vasculature.[4] In fact oxygen saturation of haemoglobin, which is dependent on lesion metabolic state and haemoglobin concentration are important diagnostic parameters to detect angiogenesis and hypermetabolism of cancers.[27]

Haemoglobin can be used for example to visualize the structure and the lesions of the brain, to delineate tumour vasculature, to monitor hemodynamic, to measure microvascular blood flow etc.

So, it is very promising in the fields of neurophysiology, neuropathology and neurotherapy. Then, it can be used to monitor some biological processes like burn recovery. In fact, it is possible to discriminate among coagulated and non-coagulated blood because of their different optical absorption spectra. However many other applications are possible.[26][23]

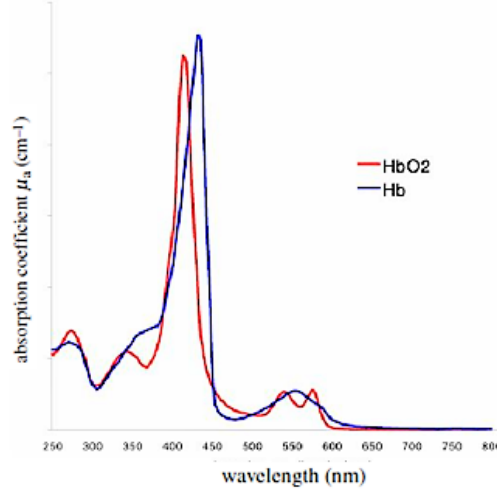


Figure 2.3: Absorption coefficient spectra of Oxyhaemoglobin (HbO_2) and deoxyhaemoglobin (Hb). [3]

2.1.3 Penetration depth

The main disadvantage of this technique is the lower penetration depth respect to ultrasound. However, this parameter is better than that of purely optical imaging.[2]

Penetration depth is limited ultimately by both optical and acoustic attenuations but in general, for most soft tissues optical attenuation dominates. It depends on absorption and scattering and, it is strongly wavelength dependent. Penetration depth is characterized by the effective attenuation coefficient μ_{eff} :

$$\mu_{eff} = (3\mu_a(\mu_a + \mu'_s))^{1/2} \quad (2.2)$$

where μ_a and μ'_s are the absorption and reduced scattering coefficients, respectively. After about 1 mm, the light becomes diffuse and the irradiance decays exponentially with depth with μ_{eff} as exponential constant. This means that, after the first millimetre, light is attenuated by a factor of 4 for each additional centimetre. In spite this, choosing the correct parameters, a penetration depth of several centimetres is possible.[2]

2.1.4 Spatial Resolution

For imaging systems, a fundamental parameter is the spatial resolution that defines the capability to distinguish two small objects with high contrast and very close together.

Spatial resolution is limited by the acoustic attenuation that is frequency dependent, so it decreases with depth. There are differences based on tissue type, but in general for centimetre penetration depth, it is sub-millimetric, for millimetre, it is under 100 μm and for some μm , it is under 10 μm . [2]

2.1.5 Ultrasounds and imaging

Ultrasounds are acoustic waves over the human audible range, so with frequency over 20 kHz. In US probes, they are generated and received using crystals with the piezoelectric effect. [28]

For imaging applications, there are two fundamental ultrasound features: velocity (v) and acoustic impedance (Z):

$$v = \lambda f \quad (2.3)$$

$$Z = \rho v \quad (2.4)$$

Where λ is the wavelength, f the frequency and ρ the tissue density. For human soft tissues the mean speed value is 1540 m/s. [28]

During their propagation in tissues, ultrasounds are attenuated depending on the attenuation coefficient α of the media:

$$A(z) = A_0 e^{-\alpha z} \quad (2.5)$$

Where A_0 is the initial amplitude.

While, passing from a material to another, there are attenuation (I), reflection (R) and transmission (T):

$$I = 20 \log_{10} \left(\frac{A_2}{A_1} \right) \quad (2.6)$$

$$R = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2} \right)^2 \quad (2.7)$$

$$T = 1 - R \quad (2.8)$$

Impedance differences among tissues must be low in order to avoid big reflection.

Ultrasounds employ a time t to do a distance d that can be measured as $d = \frac{1}{2} v \Delta t$. [28]

2.1.6 Legislation

Human exposure to EM radiations is limited by some regulations in order to not create damages to tissues. The American National Standard defined the Maximum Permissible Exposure for wavelength from 180 nm to 1 mm that is defined as the maximum level of EM radiation to which humans can be exposed without harmful effects. This value is defined in function of EM wavelength, exposure time and pulse repetition and it is expressed in J/cm² or in W/cm². [27]

For Medical Device Directive 93/42/CEE the maximum limit of exposition to electromagnetic waves is 3 W/cm² for all conditions. [29]

2.2 Prexion System

The most common sources for photoacoustic imaging are the high energy Nd:YAG lasers which can generate 4-7 ns pulses with energy of mJ. However, these systems are cumbersome, expensive and they have fluctuations of wavelength and power. Moreover, a shield is used to protect the operator from the irradiation, but it makes the system bulkier and, in this way, the operator cannot interact with the sample. This problem restricts a lot of clinical procedures. To overcome these limits, miniaturized OPOs were used but they are expensive and delicate. The possibility that is getting a foothold recently, is the use of pulse laser diodes. In fact, LEDs (light emitting diodes) are inexpensive, compact, multi-wavelength and more stable. LED-based systems are portable and in this case the enclosure and protective glasses aren't necessary.[4]

The device used in this thesis is a portable LED-based photoacoustic imaging system (PLED-PAI) that is commercially available and belongs to Prexion Corporation (Tokyo, Japan; Figure 2.4).

The excitation source are light emitting diodes characterized by high density and high power. Exactly, there are two LED arrays on either side of an ultrasound probe. Each array is composed of 4 rows of 36 single embedded LEDs. The excitation wavelength is 850 nm for two rows and 690 nm for the other two. Obviously, agents absorption peak has to match with the excitation source. In any case, several species absorb well at 690 nm or at 850 nm. The dimensions of each array are height 12.4 mm, length 86.5 mm and width 10.2 mm. The two lights emitted, collimate in a line depending on arrays inclination. Pulse width can be set up from 50 ns to 150 ns with steps of 5 ns while pulse repetition rate can be 1 kHz, 2 kHz, 3 kHz or 4 kHz and it defines consequently the temporal resolution. After each pulse, a photoacoustic signal is generated.[4]

In order to minimize the noise, it is possible to change the average among some defined numbers. This option defines the frame rate and the temporal resolution. The options are: 1 KHz (32, 64, 96, 160, 320, 640, 1600, 3200, and 6400 averages), 2 KHz (64,128,192, 320, 640,1280, 3200, 6400, and 12800 averages), 3 KHz (128, 192, 320, 640, 1280, 3200, 6400, and 12800 averages), and 4 KHz (128, 256, 384, 640, 1280, 2650, 6400,12800, and 25600 averages). So, frame rates can be 30 Hz, 15 Hz, 10 Hz, 6 Hz, 3 Hz, 1.5 Hz, 0.6 Hz, 0.3 Hz, and 0.15 Hz.[4]

For the acoustic part, there is a 128 channels ultrasound linear array transducer with central frequency that can be set up between 7 and 10 MHz that can pulse and receive. All the elements are long 3.5 mm. They make the detection in parallel so that a single frame is reconstructed. The biggest detection depth of the ultrasounds is 38 mm.[4]

It is possible to scan in a specific direction in order to acquire 3D data.

For combining images with US acquisitions, B-mode is used, where each point of the image is represented with a different intensity depending on the amplitude of the signal in that point. The different intensities are obtained with the grayscale in the bidimensional image.

Data acquisition unit is made of a dynamic range of 16 bits with 1024 samples. Photoacoustic has a sampling rate of 40 MHz, while ultrasound of 20 MHz.[4]

During the use, a slight temperature increment on the LED surface is measured because of the pulses. This leads to little LED power variations.

As it regards the characterization of the LED beam profile, 25 mm regions can be imaged with a homogenous plane of light.

With a repetition rate of 4 kHz and a pulse width of 70 ns, the power measured is 9.85 mW/cm² at 690 nm and 31.55 mW/cm² at 850 nm.

The axial resolution is 268 μ m and the lateral spatial resolution is 550-590 μ m with a pulse width of 70 nm. The best temporal resolution is 30 Hz.[4]

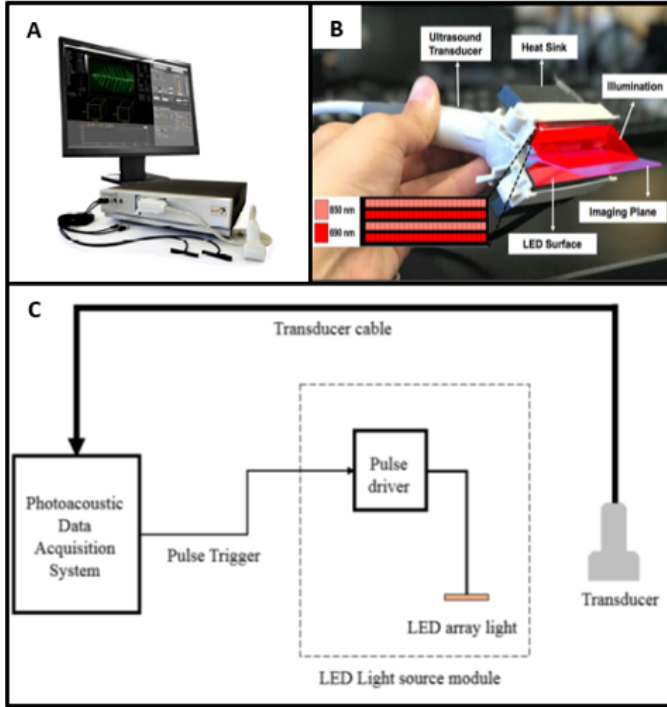


Figure 2.4: (A) PAI System of Prexion Corporation (B) PLED-PAI probe with imaging plane with LED array design in the insert (C) Scheme of PAI system.[4][5]

Chapter 3

Blood Vessels Extraction Techniques

Many different methods have been proposed for the automatic extraction of vascular network. Differences are related to pre-processing steps, computational time, accuracy and visual quality of results. The most suitable is chosen depending on imaging technique and application.[30] Among these techniques it's possible to individuate three fundamental categories: pattern recognition approaches, model-based approaches and machine learning approaches. Obviously different techniques could be used together to improve results.[11]

3.1 Pattern Recognition Techniques

Pattern recognition techniques are used for automatic detection and classification of objects. In the area of vessel extraction, they are used to detect vessel structures and features. Among these techniques, it's possible to classify different approaches:[9]

3.1.1 Multi-scale approaches (MSA)

MSA are quick and robust methods that make the segmentation at different levels of resolution. The biggest objects, so in this case largest vessels, are extracted using images with lower resolution, while smaller vessels are extracted using the ones with higher resolution, investigating the neighbourhood of the stronger structures.[9] In literature, are reported several applications, like for example 3D reconstruction of coronary arteries [31] and flow analysis of arteries with MR images. [32]

3.1.2 Skeleton-based approaches (SBA)

Skeletonization is a process to reduce objects in a binary volume to a minimal representation that keeps the morphology without redundancy. The purpose is to obtain concatenate voxels without disconnections, also in the curvatures that are the most critical points. This is very important in the case of vessel skeletonization in order to obtain the right perception of vasculature.[7]

There are two types of skeletonization techniques: *topological* and *geometric*.

The *topological* skeletonization considers the shape complexity and reduces it from outside. It uses a defined number of masks which are applied to each point of the object to establish, depending on the neighbourhood, if the conditions for the point removal are satisfied. Typically, for 3D volumes, a 26-neighbourhood is considered. From literature, it is known that the minimal number of masks necessary to obtain a skeleton without disconnections is 64. The topological technique guarantees to obtain the minimal shape, so it has high performances, but it is an iterative procedure, so there is a huge computational cost.[7]

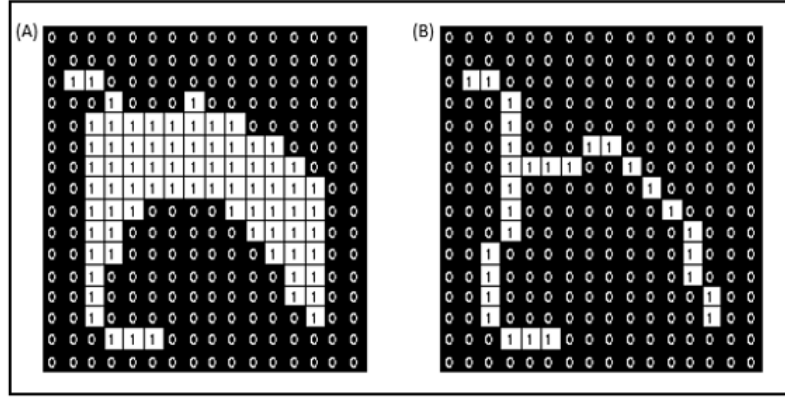


Figure 3.1: Example of thinning procedure - background black pixels and foreground white pixels: (A) Original object (B) Finished skeleton.[6]

The *geometric* skeletonization instead, doesn't consider the shape: it uses a geometric parameter. In particular, it measures the distance transform for each point of the object, that is the distance from each voxel that is 1 to the nearest that is 0, so from the edge. Then it removes the points that have a value of the distance transform under a pre-defined threshold. It does the global thresholding of the distance transform. So, threshold choice is fundamental, because if it is too high disconnections are created, while if it is too low the result isn't the minimal skeleton.[7]

$$Sk = Dst \text{ for } (Dst > tp) \quad (3.1)$$

where Sk : Skeleton points, Dst : matrix with Distance transform for each point, tp : parameter for the definition of skeleton thickness (threshold).

The geometric procedure has lower computational cost but less good performances.[7]

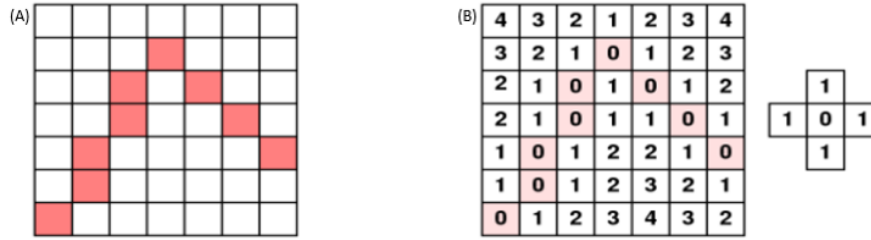


Figure 3.2: Example of Distance Transform: (A) Object indicated in coloured pixels (B) Distance Transform calculated with the mask on right (moving only on rows and columns).[7]

It is possible to use these two methods in cascade: the geometric with a low threshold before, in order to remove the majority of the points and then the topological to obtain the minimal representation.[7]

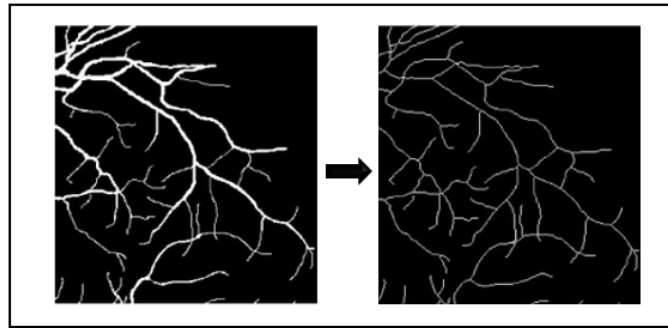


Figure 3.3: Example of vessels skeletonization: on the left the starting mask and on the right the final skeleton.[8]

Even in this case there are a lot of practical applications, for example vessels extraction from CT images of the lung using a thinning algorithm, in order to detect tumours.[33]

3.1.3 Ridge-based approaches (RBA)

RBA consider grayscale images as 3D elevation maps where the skeleton of the tubular objects is approximated with intensity ridges. Ridge points are defined as the local peaks in the maximum surface gradient direction and can be obtained by tracing elevation maps from an arbitrary point along the steepest ascent direction. This approach is applicable to different imaging techniques and it doesn't depend on affine transformations.[9] In literature, there are several applications, such as for the time observation of pathologies evolution.[34]

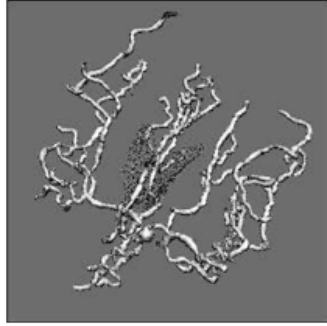


Figure 3.4: Vessel tree extracted from mouse clicks.[9]

3.1.4 Region Growing approaches

In this procedure some seed points are selected. Then starting from them, pixels are recruited exploring neighbourhood by using some predefined criteria like value similarity or spatial proximity. The assumption is that pixels close to each other and with similar intensity, belong to the same object. Because of intensity variations and noise, some errors are possible, such as holes and over-segmentation, therefore postprocessing is necessary. Among the possible uses, there is the segmentation of coronary vessels.[9]

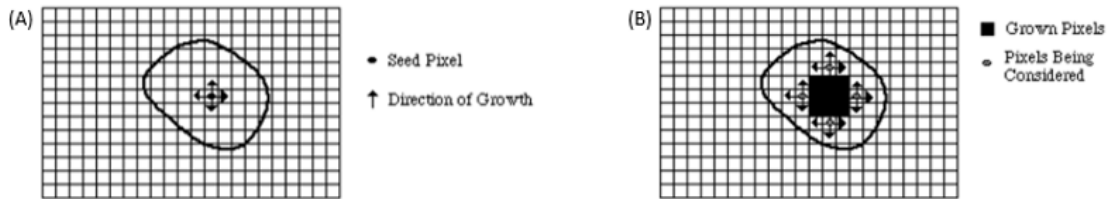


Figure 3.5: Region Growing procedure: (A) Start of growing region (B) Growing process after few iterations.[7]

3.1.5 Differential Geometry-based Approaches (DG)

DG treats images as hypersurfaces and extract features exploiting curvature and surface crest lines that are center lines. For example, 2D and 3D images are modelled as 3D and 4D hypersurfaces respectively. These features are invariant to affine transformations. Principal curvatures are defined as the eigenvalues of the Weingarten matrix, while principal directions are the eigenvectors. Crest points are the local maxima of the curvature in the hypersurface and center-lines are the links between crest points.[9]

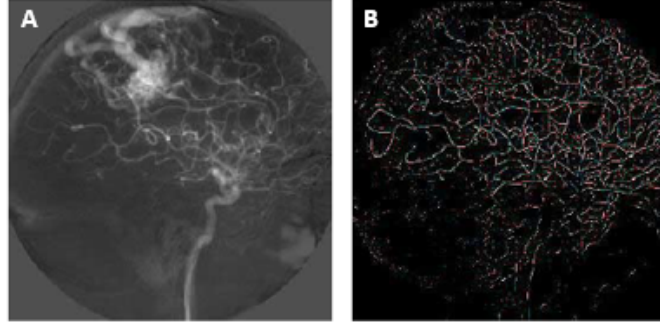


Figure 3.6: (A) DSA image of the cerebral vessels (B) Vessel detection using four scales.[9]

3.1.6 Matching Filters Approaches

They apply the convolution of the image with multiple matched filters in order to extract the object of interest. Obviously for extracting vessel contours, are designed different filters to detect vessels with various orientation and sizes. The size of the kernel influences the computational time.[9] The convolution among the kernel and the image is:

$$g[m,n] = h[k,l] * a[m,n] = \sum_{k=-K}^K \sum_{l=-L}^L h[k,l]a[m-k,n-l] \quad (3.2)$$

Where g is the image after filtering, a is the original image and h is the kernel.

The convolution is performed with the multiplication of every kernel value by the corresponding pixel in the input image.[7]

Among the possible filters, there is the median filter.

Median filter is a nonlinear filter widely used in image processing because of its great efficiency in the removal of “salt and pepper noise”. This noise is referred to the image outliers that can occur for different reasons, such as bit errors in transmission or in the procedure of signal acquisition. For a correct operation, it’s important that the number of black and white pixels is similar.

This filter works replacing the current pixel with the median value of pixels in a predefined neighbourhood. The kernel size is an important parameter for the determination of the median value, in fact, in general, the appropriate size isn’t large, or the value deviates too much from the correct one. In the case of an even number of pixels, the average of the two median values is considered. Median filter is preferred compared to mean filter because it allows to preserve objects boundaries.[35]

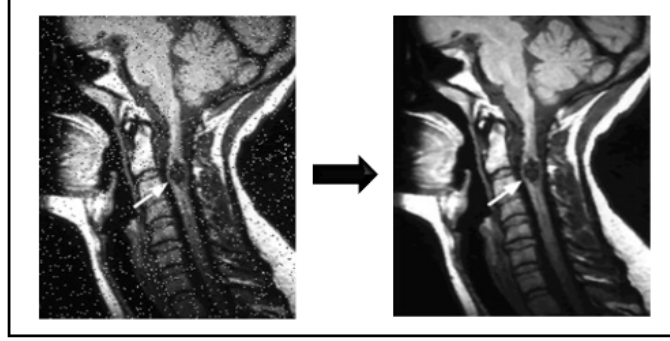


Figure 3.7: Corrupted image with salt and pepper noise on the left, enhanced image with a 3 by 3 median filter on the right.[10]

3.1.7 Global Thresholding

The simplest segmentation method is the global thresholding. It can be applied if the luminance histogram is bi-modal and it consists in the choice of a threshold for the separation of foreground and background pixels.[9]

$$\begin{cases} 1, & \text{if } a[m,n] > T \\ 0, & \text{if } a[m,n] \leq T \end{cases} \quad (3.3)$$

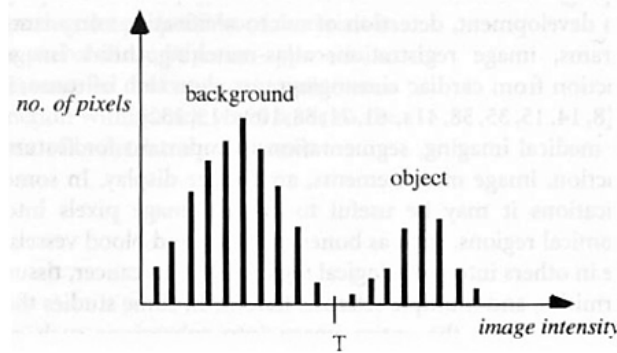


Figure 3.8: Bi-modal luminance histogram with separation of background and object.[7]

The Otsu method can be used in order to establish the threshold automatically, so that the intra-class variance is minimized, or symmetrically the inter-class variance is maximized.[36]

Let the pixels of the image be represented in L grey levels. At each level i the number of pixels is

$$n_i, \text{ while the total number of pixels is } N = \sum_{i=1}^L n_i$$

The intra-class variance is defined as the weighted sum of the two classes variances:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (3.4)$$

Where t is the threshold, ω_0 and ω_1 are the two classes probabilities and σ_0^2 and σ_1^2 are the two classes variances.

The probabilities are calculated from the L histogram bins:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \text{ and } \omega_1(t) = \sum_{i=t}^{L-1} p(i)$$

Minimizing the intra-class variance is equivalent to maximize the inter-class variance:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \quad (3.5)$$

Where the class means are: $\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)}$, $\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)}$, $\mu_T = \sum_{i=0}^{L-1} ip(i)$

So, it is demonstrated that $\omega_0\mu_0 + \omega_1\mu_1 = \mu_T$ and $\omega_0 + \omega_1 = 1$.

The algorithm sets up the initial $\omega_i(0)$ and $\mu_i(0)$ and then it iterates from $t = 1$ to $t = \text{maximum intensity}$, updating the values of ω_i and μ_i and computing $\sigma_b^2(t)$ for each iteration. The maximum among the $\sigma_b^2(t)$ calculated is the threshold.[37][36]

To work, this method needs deep and sharp valleys between peaks. However, in real images, valley bottom detection isn't so simple. Some factors can affect the eligibility of the histogram for the threshold choice, for example if the object is very small respect the background because there isn't bi-modality, or if there is noise, or the variances of the objects and the background are too large, or if there is small means difference.[36][38]

There are a lot of examples in literature, such as the segmentation of brain vessels from MRI or liver vessels from abdominal CT and from retinal images.[39][40]

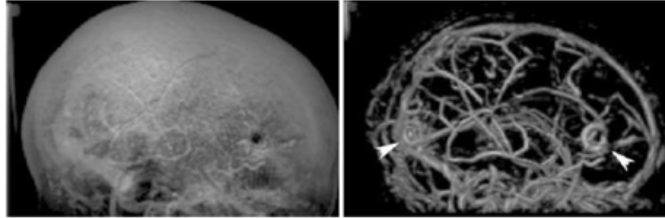


Figure 3.9: Original volume on the left and filtered MR images on the right.[9]

3.1.8 Mathematical Morphology Schemes

Morphological operators are another technique introduced in order to extract specific morphologies from the image. They can be used in black and white or monochromatic images, but their principal use is with the first, for the creation of masks where some imperfections have to be deleted.

For the application of the morphological operators, it's important to define the structuring element that is a connected around with specific dimension and shape. The structuring element defines the operator locality. There are several possible geometries, depending on the image morphology,

such as line, circle, square, diamond, ring and others.

The principal operators are Dilatation and Erosion. In the erosion, the output value is the minimum among the pixels selected from the structuring element, while in the dilatation is the maximum. So, with the former, the boundaries of foreground pixels are eroded, while with the latter they are enlarged.

Combining these two operators, others can be realised. The most important are the opening and the closing. The first is the erosion followed by the dilatation with the same structuring element and the second the opposite. The opening removes objects smaller than structuring element preserving background, the closing is useful to do “hole filling”. [7]

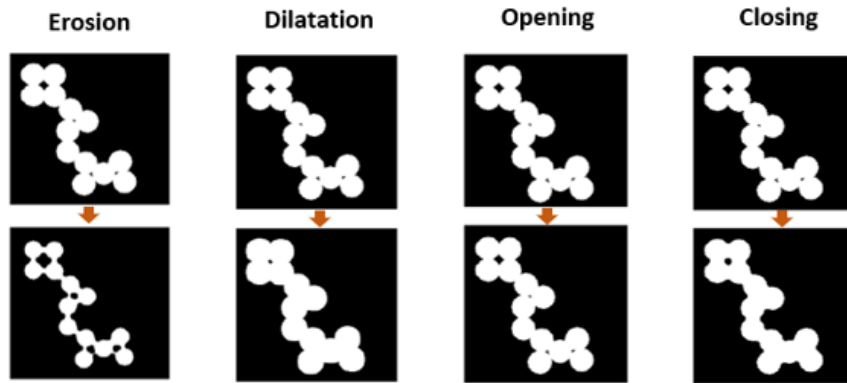


Figure 3.10: Example of morphological operations using a disk with ray 5. [7]

With these methods, it's been possible for example to detect coronary arteries in cine-angiogram. [41]

3.2 Model-based Techniques

Model-based approaches apply to the images, explicit vessel models with the intent to extract vasculature. They can be divided in three different groups: deformable models, parametric models and template matching. [9]

3.2.1 Deformable Models (DM)

DM take in consideration curves or surfaces of the image that can be deformed by the influence of internal and external forces. Internal forces are important for the smoothness of the surface, while external ones attract it to the vessel boundary. The smoothness constraint is the elasticity energy and makes the DM robust to the noise. In order to start the process, the surface has to be initialised and it's important that the model is enough robust to be insensitive to initialization and noise. With recent implementations it's possible to insert constraints or a priori knowledge about geometry. These approaches are suitable for complex architecture or variable vessels, but they are very time consuming. They are divided in parametric and geometric. [9][7][11]

For *Parametric* deformable models the formulation of the problem is: $c = \frac{\partial S}{\partial t} = F_{int}(S) + F_{ext}(S)$

where c is a damping coefficient and $F_{int} = \frac{\partial}{\partial s}(p \frac{\partial S}{\partial s}) - \frac{\partial^2}{\partial s^2}(q \frac{\partial^2 S}{\partial s^2})$

s is included between 0 and 1, while p and q are the parameters to control the elasticity and the resistance to deformation of the surface. F_{ext} is different depending on method. With respect to geometric models, they have more difficult to topological changes adaptation, but they are easy and fast.[9][7][11]

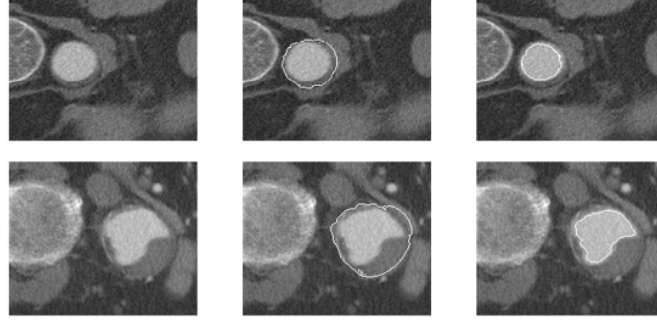


Figure 3.11: Segmentation of thoracic aorta with a parametric deformable model.[7]

Snakes are a special type of parametric models, where the active contour allows to minimize energy. The disadvantage is that the user has to set connected control points, called snaxels. Each of them, has an associated energy that can grow or fall depending on the forces at stake. Following implementations of snakes overcome traditional limits such as the initialization of contour and parameters. The problems are that it's necessary to initialise snake near to the edge of interest and that they can't follow concavities of figures.[9][7]

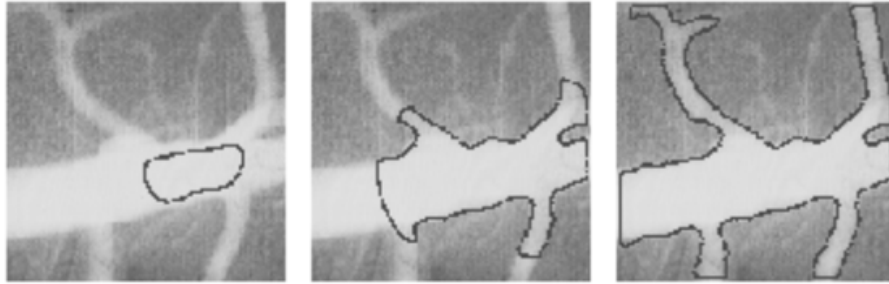


Figure 3.12: Segmentation of retinal vessel with a snake.[7]

Geometric deformable models are based on the theory of curve-evolution: $\frac{\partial S}{\partial t} = Fn$ Where F is the speed function and n is the unit normal to S . Speed function depend on a constant term and another that depend on the curvatures: $F = F_{attr} + \beta F_{curv}$ With this method it's easy to handle topological changes because the evolution of the surface is dependent only on geometric constraints. It is independent from initial conditions. A common implementation is the Level Set: $\frac{\partial \Phi}{\partial t} = Z(\zeta)|\nabla(\Phi)|$ LS represents propagating curves as the zero level of a function at higher dimension in the Eulerian system. In this way, it's possible to handle complex interfaces.

Varying the K level of the contour or the level set function in the time, the contour is dynamic and follows the shapes in the image.[9][7]

$$\begin{cases} \Phi(x,y,t) > 0, & \text{for } (x,y) \in \Omega_o \\ \Phi(x,y,t) < 0, & \text{for } (x,y) \in \Omega_i \\ \Phi(x,y,t) = 0, & \text{for } (x,y) \in C(x,y,t) \end{cases} \quad (3.6)$$

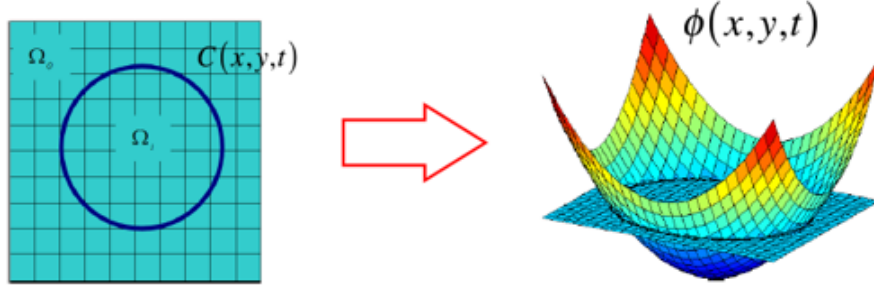


Figure 3.13: Level set implementation: function in the higher dimensional space on the right and its K level on the left.[7]

3.2.2 Parametric Models (PM)

They define parametrically the object of interest. In particular, for tubular objects, they are described as a set of overlapping ellipsoids. In some applications, the model of the vessel is circular. The estimation of parameters is done from the image, but the elliptic PM approximates well healthy vessels but not pathological shapes and bifurcations.[9] With this approach for example, it was possible to find diameters of small vessels in X-ray cine-angiograms using information in intensity profile amplitude.[42]

3.2.3 Template Matching

This method attempts to recognize a structure a priori model (template) in the image. This is a contextual top-down method. For the application of arterial extraction, the template is a set of nodes connected in segments, that than is deformed to fit the real structure. For the deformation, stochastic process can be used.[9] This technique can be used for determinate flow direction and vessel diameter.[43]

3.3 Machine Learning

Machine learning is a subfield of artificial intelligence in which computers learn how to solve a specific problem from experimental data.[12]

These approaches can be divided in unsupervised and supervised:

- Unsupervised approaches try to find a model that describes input images no having prior knowledge about them. This technique doesn't need the comparison with gold standard, so it is very useful when GS is not available.
- Supervised learns the model from a training set of labeled images and then applies it to the input images. This technique has better performances, training computational cost is huge, but test is very fast. In particular, for each element there is a vector with features and the desired output value.[11]

They exploit knowledge in order to guide segmentation of vessel structures. A possible knowledge is the type of image, sometimes also general blood vessel models can be used as knowledge. These methods in literature are used for example in cardiac angiograms.[44]

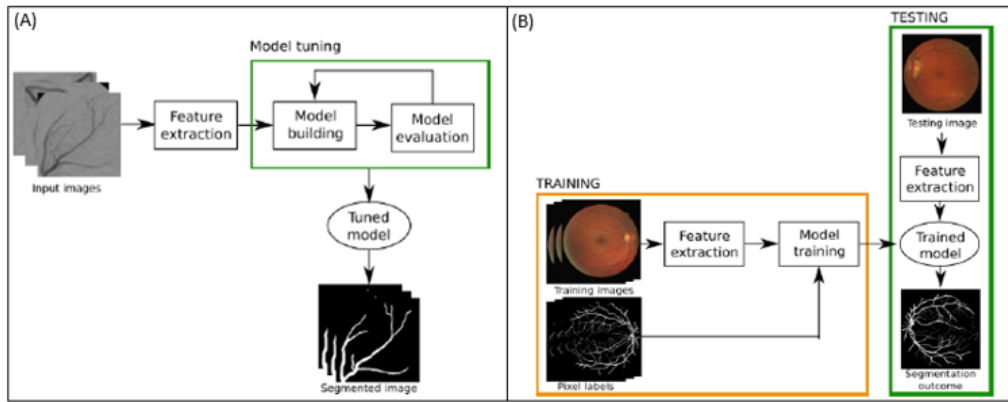


Figure 3.14: (A) Scheme of Unsupervised learning approach (B) Scheme of Supervised learning approach.[11]

Machine learning field is in continuous expansion and a lot of techniques have been proposed. Among them there are Neural Networks: NN are inspired by the way in which brain performs a particular learning task. The network is composed by nodes (neurons), that are the elementary processors connected by links (synapsis). Each node has a number of inputs, makes some computation and generates an output. Each link has a weight and the output of every neuron is a function of the weighted sum of the inputs. Weights are learned during training for being use in recognition. Among learning algorithm, one very used is back-propagation algorithm.[9][12]

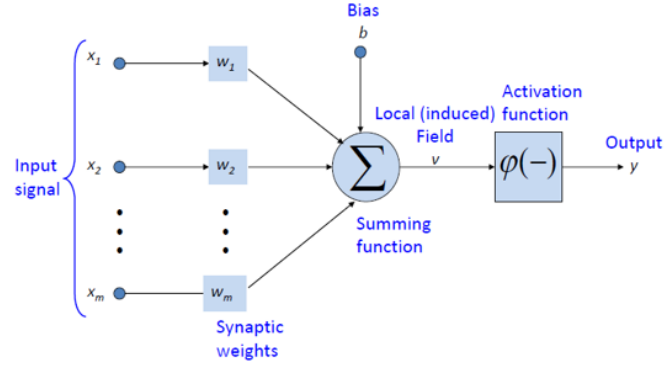


Figure 3.15: Model of a neuron.[12]

Recently, Convolutional Neural Networks CNNs are developed. They are artificial neural networks inspired by the human visual cortex. They are characterized by the presence of Convolutional layers for feature extraction, Pooling layers for features reduction and fully connected layers for classification.[11][12]

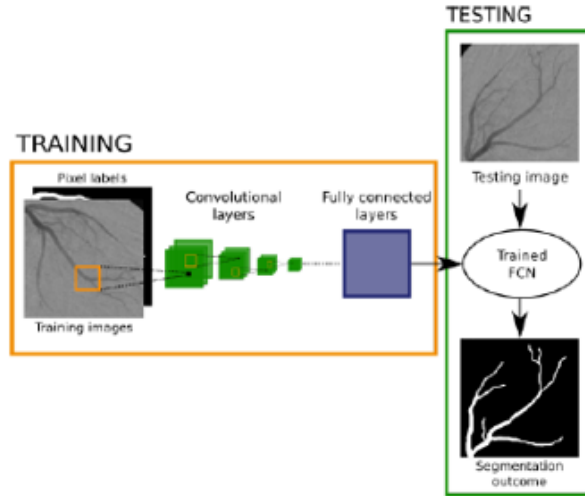


Figure 3.16: Convolutional Neural Network for vascular segmentation.[11]

Chapter 4

Phantoms

4.1 Phantoms importance in biomedical imaging

In medical imaging, phantoms are samples with known geometry and composition that mimic biological tissues with their physical and chemical properties for providing a realistic environment for clinical imaging applications. Stable and well characterized phantoms are very useful for doing routine quality controls, training, calibration and for evaluating performances of systems and algorithms. They can be used also for new applications development before to use animals or humans. Moreover, phantoms allow to understand reproducibility in time and among laboratories, to optimize signal to noise ratio, to compare detection limits and accuracies of different systems and to examine maximum possible depth.[45][46][47][48]

PAI phantoms should have optical and acoustic properties of the tissue of interest. In particular, the most important properties are optical absorption, optical scattering, speed of sound and acoustic attenuation. For blood, they are: absorption coefficient $2\text{-}10\text{ cm}^{-1}$, reduced scattering coefficient $10\text{-}15\text{ cm}^{-1}$, speed of sound 1560 m/s and acoustic attenuation coefficient $0.1\text{-}2\text{ dB/cm}$. [45][48]

With this purpose, a lot of materials were indagated, such as hydrogels, paraffin, polymers, epoxies and liquid fat emulsions, also with the addition of optical absorbers and scatters. However, polymers and elastomers have a different speed of sounds compared to tissues, instead liquid fat emulsions have a low acoustic attenuation, a limited range of sound speed and suffer of degradation. The problems of hydrogels such as agarose and gelatine are that they can be affected by fungal infections, bacterial growth, air absorption, dehydration, physical damage and mechanical degradation, so they aren't suitable for long term applications. Epoxy resin and polyester have optical transparency and long-term stability, but high sound attenuation. Another possibility is polyvinyl alcohol PVA that has good optical and acoustic properties but for its preparation needs a complex process of freeze-thaw cycles and it is sensible to humidity. Finally, there are copolymer-in-oil gels like polyvinyl chloride plastisol (PVCP) that have high temporal stability but also too high acoustic attenuation.[45][46][49][47]

It's possible to increase water-based materials longevity with care in storage and adding chemical substances such as formaldehyde. In any case, soluble dyes diffuse in water so it's necessary some encapsulation, for example in PMMA tubes. Agarose and gelatine are much used because they are cheap, well characterized and easy to fabricate and to modify with other particles.[47]

The possible absorbers are inks, fluorescent dyes and nanoparticles.[49] While optical scattering can be obtained with intralipid and titanium oxide and acoustic scattering with silica particles.[47]

In this work, hydrogels were chosen for their low cost and fabrication simplicity and because phantoms use time was short so there weren't problems with storage. As absorber, an ink was used.

4.2 Phantoms design and fabrication

4.2.1 Solidworks design

Three different phantoms were created. The first, with a little number of vessels, is the simplest (M_{simple}), the second is more complex ($M_{complex}$) and the third is deeper and it has vessels with variable diameters (M_{deep}).

Vessels were designed with the CAD software *Solidworks* and then realised with the *3D printer ProJet MJP 2500 Plus*.

In *Solidworks*, the structure of the vessels was done with the function *3D Sketch*, selecting among sketch entities, in the toolbar, the necessary instruments such as *line*, *circle*, *plane* and in order to trim away the corners at the intersections, *sketch fillets*.

Then, the hollow tubes were created from the path using the function *swept boss* and selecting the options *circular profile* and *thin feature*. In this step, for each vessel, internal diameter and wall thickness are defined.

For the third model, variable diameters were obtained using the function *lofted boss* and the option *thin feature*, selecting for each vessel the two circular profiles to connect. The two circles are done with two parallel planes with a precise distance.

Finally, for removing the intersected objects inside the tubes, to obtain a perfect hollow model, *swept cut* and *lofted cut* are used.

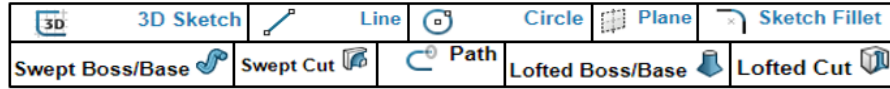


Figure 4.1: Principal Solidworks functions used.

Each model was realised with two different wall thicknesses: 1 mm and 500 μm . Then only one thickness for model was chosen depending on post-printing quality.

The three models designed have the following measures:

M_{simple} : 39,2232 mm x 34,0365 mm x 8,9184 mm - wall thickness: 1mm.

$M_{complex}$: 39,2260 mm x 34,4653 mm x 12,7786 mm - wall thickness: 1mm.

M_{deep} : 40,3752 mm x 38,1032 mm x 31,1138 mm - wall thickness: 500 μm .

In order to print them, models were saved in STL file format.

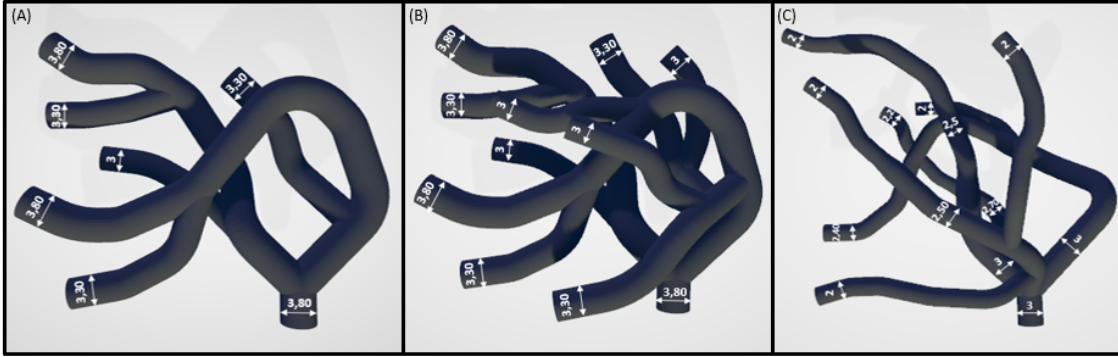


Figure 4.2: Models with the indication of external diameters - (A) M_{simple} (B) $M_{complex}$ (C) M_{deep} .

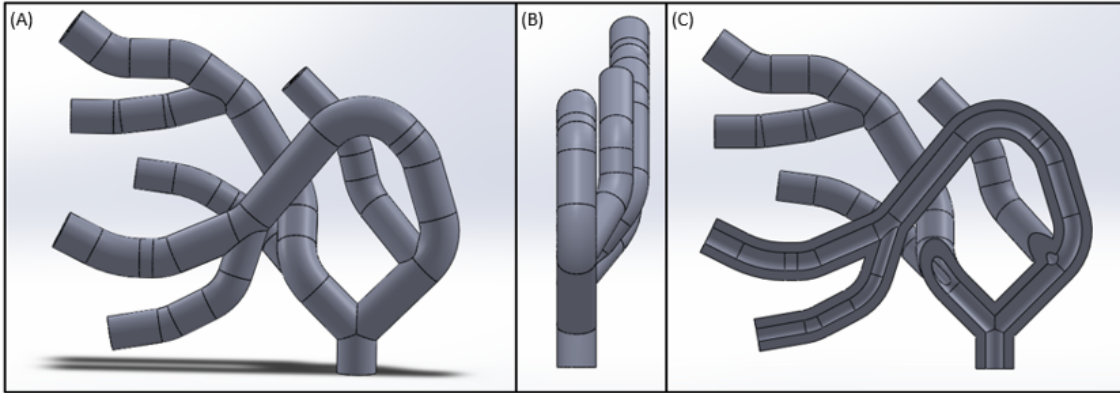


Figure 4.3: M_{simple} in Solidworks - (A) Front view (B) Lateral view (C) Section view.

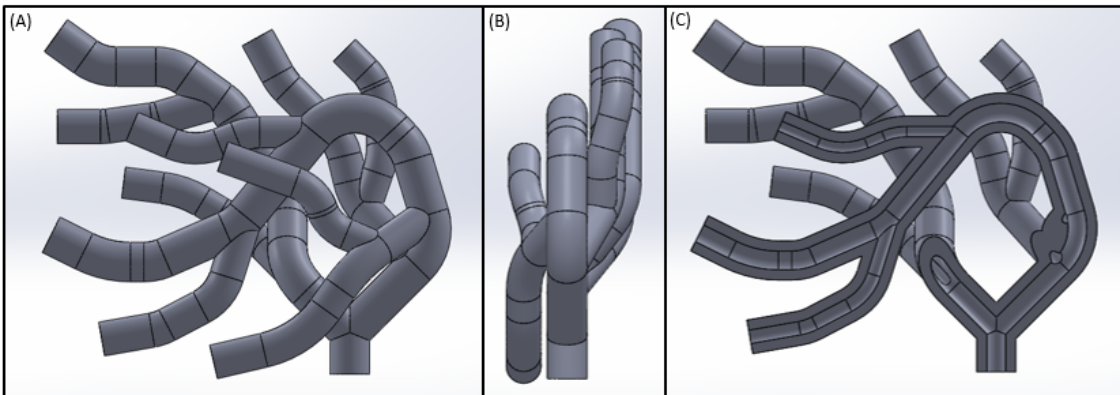


Figure 4.4: $M_{complex}$ in Solidworks - (A) Front view (B) Lateral view (C) Section view.

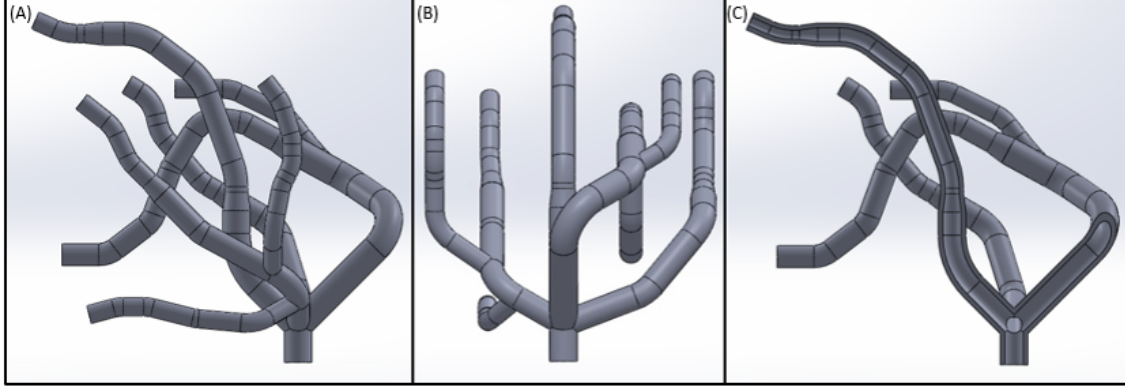


Figure 4.5: M_{deep} in Solidworks - (A) Front view (B) Lateral view (C) Section view.

4.2.2 3D Printing

ProJet MJP 2500 Plus is a 3D Multijet printer that uses the inkjet printing process. In particular, a piezo printhead deposits a plastic resin and a casting wax material trough the layer by layer technique.

Then the *MJP EasyClean System* is used to remove in a little time, the support material from plastic parts using steam and EZ Rinse-C. It is composed of two warmer units, one for bulk wax removal and one for fine wax removal. The support material is separated by melting or dissolving. This is a non-contact method, so there are less substrate or mask damages and contamination. Moreover it has high resolution (32μ) and it is economical.

The material used is VisiJet[®] Armor (M2G-CL), a tough, ABS-like clear plastic that combines tensile strength and flex.[13][50]

For printing, in the deepest model some supports are inserted among vessels in order to avoid collapse.

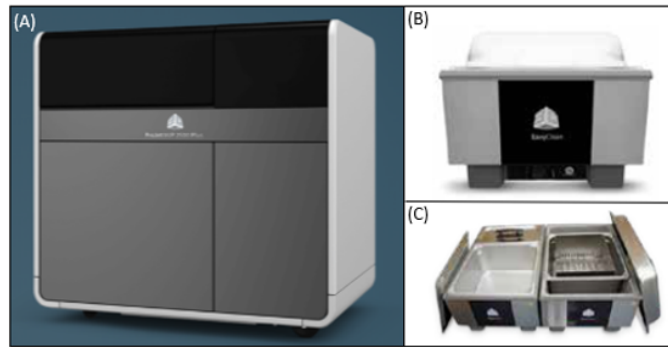


Figure 4.6: (A) Project MJP 2500 Plus (B) MJP EasyClean System (C) MJP EasyClean System open. [13]

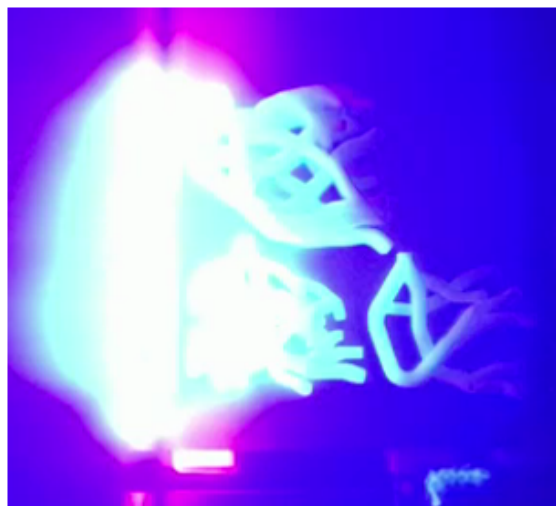


Figure 4.7: Photograph of the process of printing.



Figure 4.8: Printed models before wax removal - (A) Top view (B) Front view.

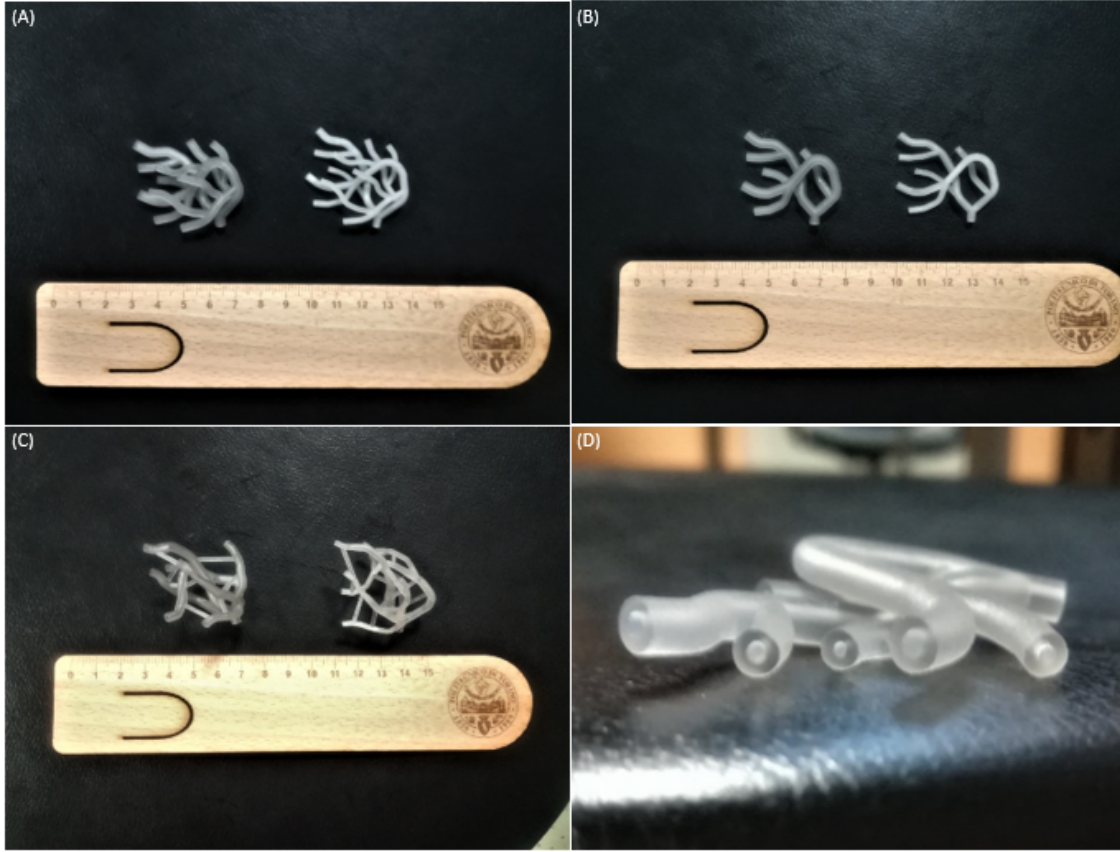


Figure 4.9: Printed models - (A) $M_{complex}$ (B) M_{simple} (C) M_{deep} (D) Zoom on M_{simple} .

4.2.3 Phantoms Realisation

The printed models were filled with a black paint of Scribbles[®] using a syringe. Black is suitable for this application because it absorbs at the wavelength of Prexion. Moreover, this paint is sufficiently dense, so it doesn't come out from tubes. Before of choosing this ink, a test was done in order to verify that it is visible with PAI. As it is shown in figure 4.10, a phantom was created inserting four tubes filled with some inks in the agar gel and some acquisition were made. Differently from green ones, black inks generate a good signal. In particular, the third from left was chosen.

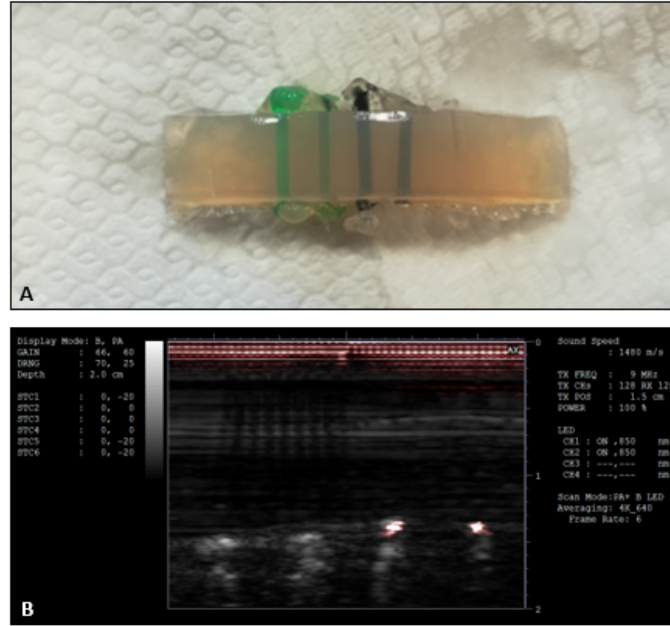


Figure 4.10: Ink test - (A) Phantoms with four different inks above (B) corresponding acquired image below.

Because of the narrow time available, not all the wax was removed from vessels inner channel. This was the principal limitation of this thesis because the ink didn't fill all the channels and so some vessels were invisible to photoacoustic imaging.



Figure 4.11: Models filled with black ink.

For the realisation of the phantoms, agar agar was chosen. It is a jellying polysaccharide, obtained from red algae and it is used to prepare transparent and neutral gels. Agar agar powder dissolves around at 90-100 °C and it solidifies at 45 °C. The dose for 1 kg of solution, is 7-10 g of powder.[51]

The desired quantity of agar agar powder was weighed with a digital balance and then it was put

in a small pot with the corresponding quantity of water stirring at the same time. The obtained solution was brought slowly to the boil with a burner continuing to stir and at this point, the warm solution was poured in a container with the vessel model. The phantom was let to cool down and when it became solid, it was pulled out from the container. (Figure 4.12)

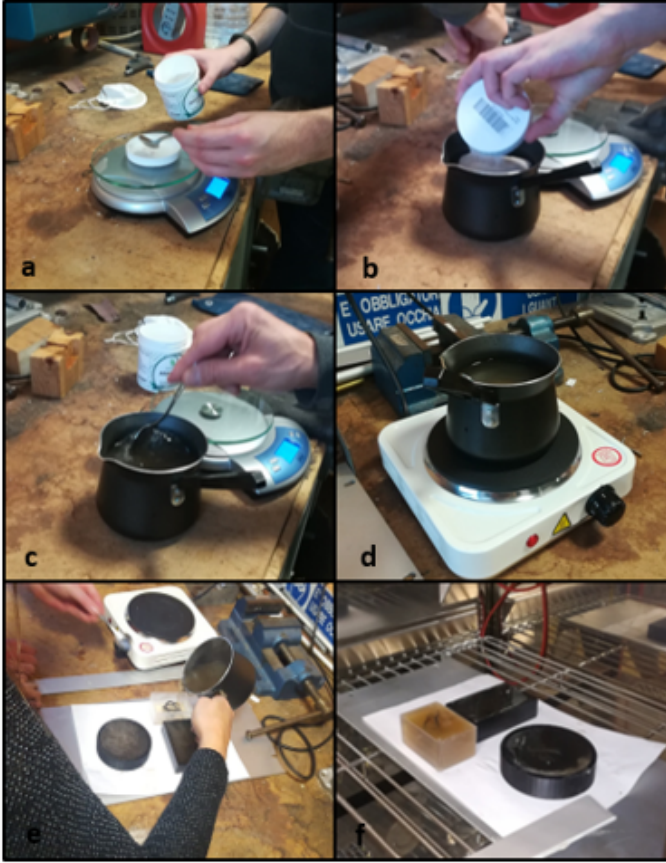


Figure 4.12: Agar preparation steps
- (a) Agar weighted with a digital balance (b) Powder put in a small pot with the water (c) Stirring (d) Solution is brought to the boil (e) Solution pouring in a container (f) Cooling down.[4][5]

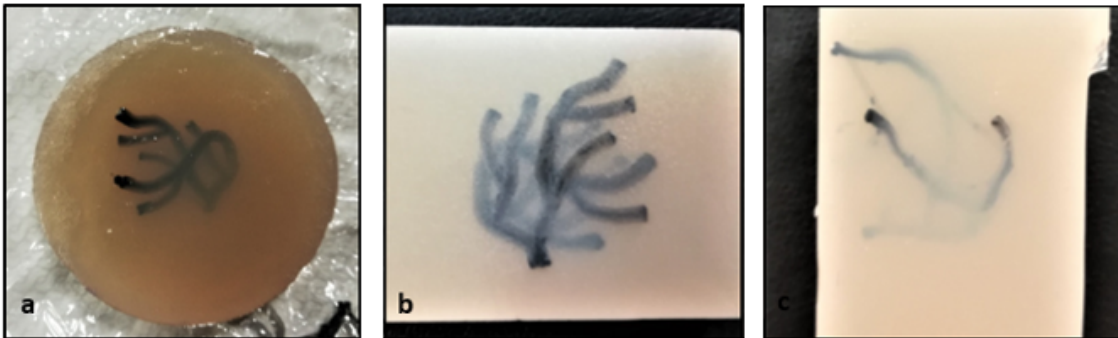


Figure 4.13: Finished phantoms - (a) M_{simple} (b) $M_{complex}$ (c) M_{deep} .

Chapter 5

Data Acquisition and Processing

5.1 Images Acquisition

5.1.1 Setup for the acquisition

For the acquisition, the phantoms were fixed to the base of a transparent container full of water. Ultrasound probe and LEDs arrays were secured to a metallic angle beam, which in turn was fixed to a mobile support that moves along a binary in response to a knob rotation.

Probe and LEDs were put underwater near the phantom and linear scansion was made moving the system one mm at a time. For each position was acquired a slice of the phantom, stopping the scansion and clicking on the buttons “Save Image” and “Save Screen”. For the three models 50, 64 and 47 slices were obtained respectively.

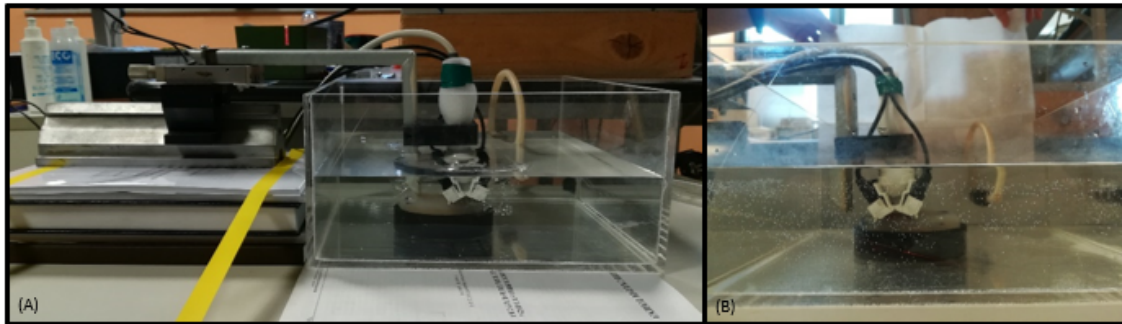


Figure 5.1: (A) Setup for images acquisition (B) Zoom on probe and phantom.

5.1.2 Prexion settings

For the acquisition Scan mode was set up to PA + B Mode while Display Mode to PA. Depth displayed was 3 cm. The frame rate was 6 Hz. As it regards LEDs, Channel 1 and Channel 2 were used with a wavelength of 850 nm. While for ultrasound probe, the 128 transducers were used with a central frequency of 9 MHz. Sound speed was set up to 1480 m/s and the pulse repetition

frequency was 4 kHz with a 640 averaging.

The following steps of Image segmentation, skeletonization and extraction of tortuosity parameters were performed with MATLAB[®].

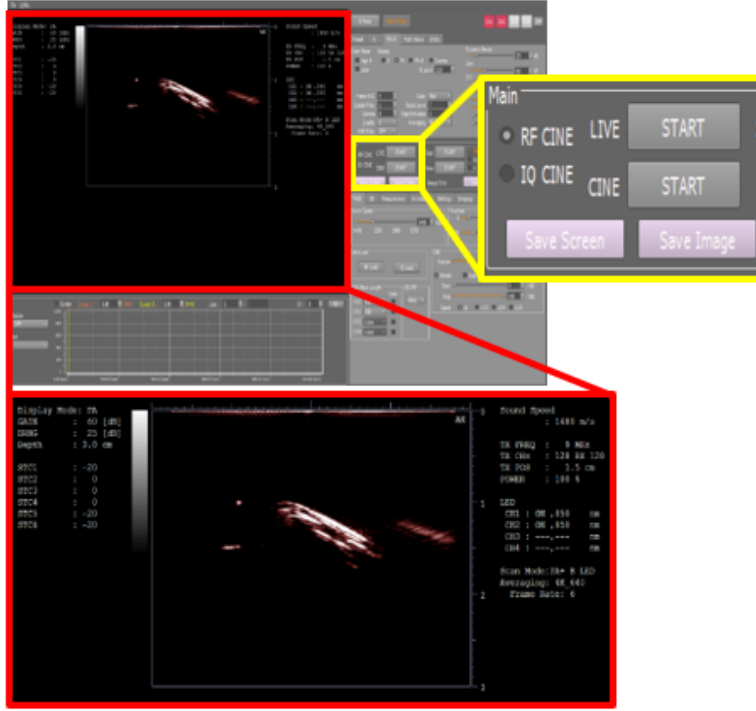


Figure 5.2: Interface for the visualization and the acquisition of images, in yellow the buttons for the acquisition and in red the screen with the settings.

5.2 Segmentation

5.2.1 Automatic Segmentation

After the acquisition, it's necessary to obtain the binary volume of the three models before making the skeletonization. So, for each of them, image processing and segmentation were performed as described below. Firstly, every acquired slice was read with the function *imread* and converted from RGB to grayscale with the function *rgb2gray*.

Then, they were cropped for deleting information displayed such as device settings, colorbar and scale bar, maintaining only the image acquired.

In particular, each slice was kept from row 46 to row 502 and from column 282 to column 889.

The obtained images were inserted in a 3D matrix with the same number of rows and of columns of the slices and the third size equal to the number of slices.

So, for the three models, the sizes are:

M_{simple} : 457-by-608-by-50, $M_{complex}$: 457-by-608-by-64, M_{deep} : 457-by-608-by-47.

Data in the matrix were stored as 1-byte unsigned integers (*uint8*).

Then, a 3D median filter was applied to the volume, using the function *medfilt3* with a 3-by-3-by-3 kernel and padding the image by repeating border elements in a mirrored way.

Later, in order to fill vessels where only walls are visible, the function *imclose* was used to do a closing operation with a disk-shaped structuring element of radius 5.

For the segmentation, the Otsu method was used to find the global threshold of each slice and then, the maximum among these was chosen to define a unique threshold for all the slices of the model. In this step, the function used is *graythresh* for which the matrix was converted in double data type with the function *im2double*.

Using the defined threshold, the masks of all the slices were obtained putting to 1 the pixels with bigger value and to 0 the pixels with smaller value.

Values of threshold for the three models are: M_{simple} : 0.4196, $M_{complex}$: 0.4275, M_{deep} : 0.3922.

Each mask was processed removing all the objects with area smaller to the 2% of the biggest object in the mask using *bwareaopen*, applying dilatation with a disk-shaped structuring element with radius 3 (*imdilate*) and erosion with a disk-shaped structuring element with radius 1 (*imerode*) and filling holes with the function *imfill*. For each slice, the binary matrix of the mask, its figure and the original image with the segmented boundary in red, were saved.

Finally, all the masks were inserted in a matrix BW. Also this matrix was saved.

The program ImageJ[®] was used in order to reconstruct the volumes in AVI video format starting from masks. Images were selected clicking on *Import - Image Sequence* among *File* options. Thereafter videos were created using *Image* options: *Stacks - 3D Projects* where some parameters can be chosen, such as the axis of rotation and slice spacing in pixels.

Comparing reconstruction of original images and reconstruction of obtained masks, it is clear that the disconnections among vessels are due to disconnections in the original images for the ink lack.

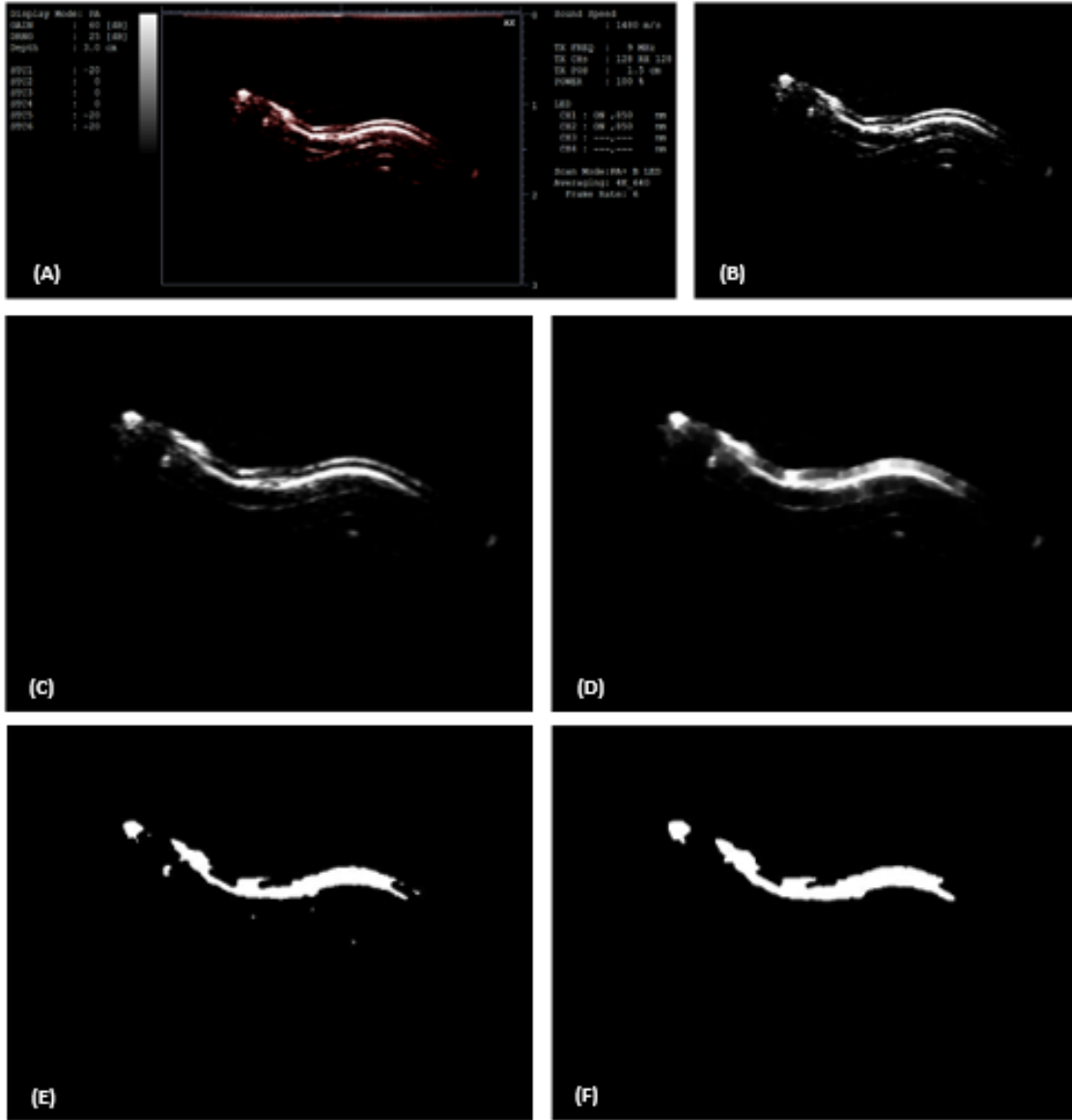


Figure 5.3: Principal steps of automatic segmentation - (A) Original image (B) Cropped grayscale image (C) Median filtered image (D) Closed image (E) Thresholded image (F) Final mask.

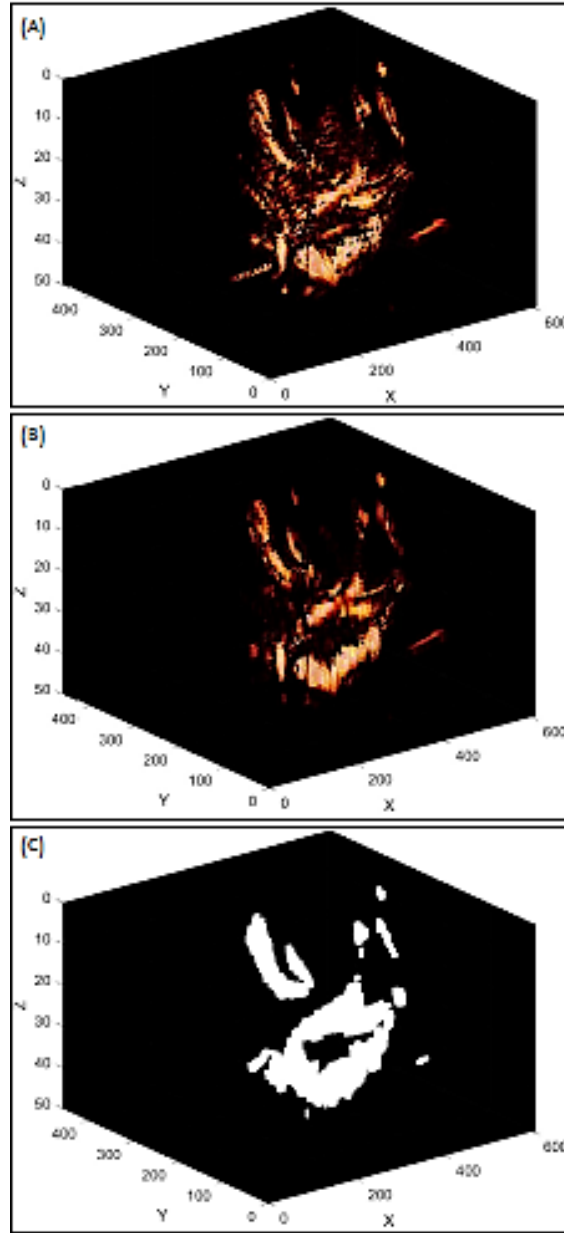


Figure 5.4: M_{simple} : (A) Original model (B) model after median filter (C) segmented model.

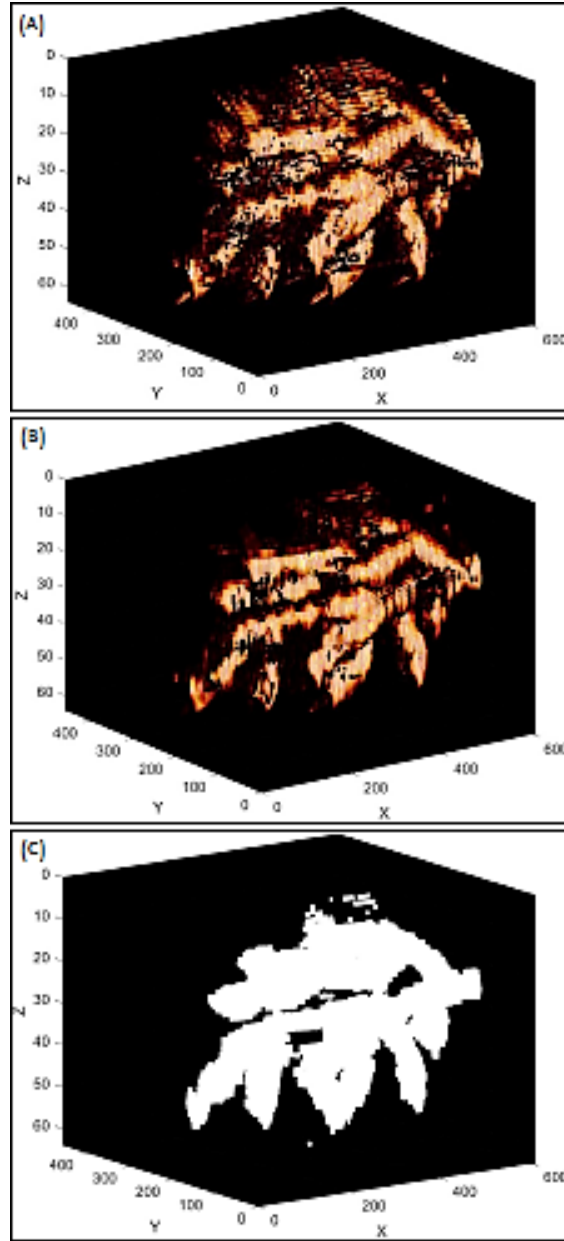


Figure 5.5: $M_{complex}$: (A) Original model (B) model after median filter (C) segmented model.

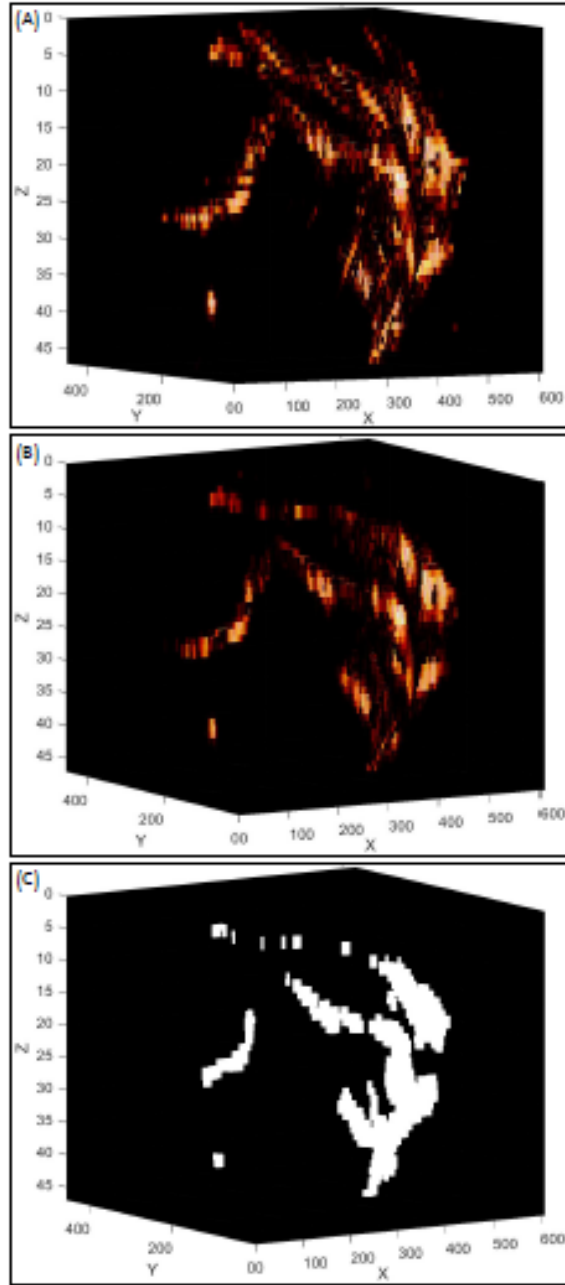


Figure 5.6: M_{deep} : (A) Original model (B) model after median filter (C) segmented model.

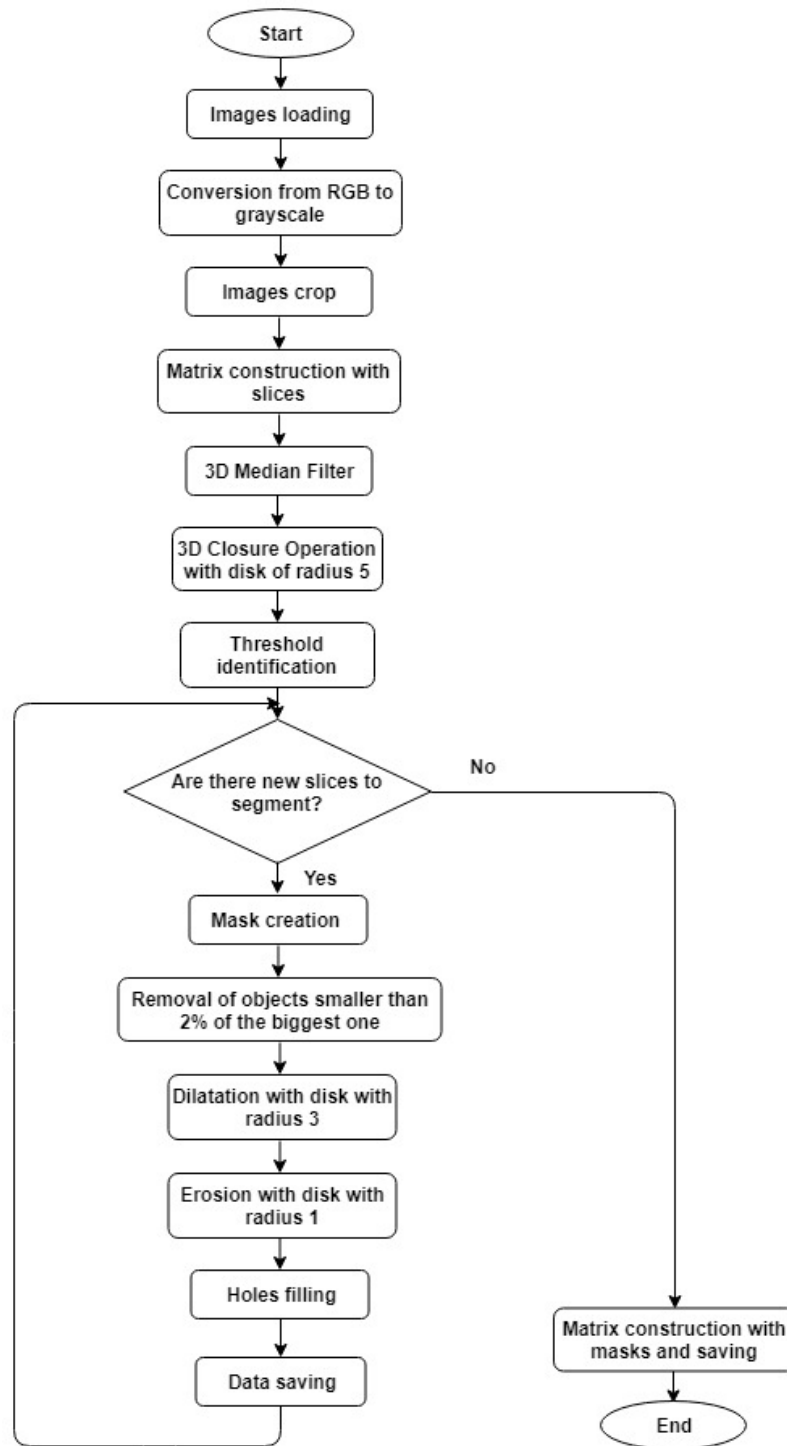


Figure 5.7: Automatic Segmentation Flowchart.

5.2.2 Manual Segmentation

Manual segmentation was done in order to validate the automatic algorithm implemented. The original images contain a lot of noise that it's difficult to discriminate from the signal with naked eye. Therefore, it was decided to segment the images after the application of the median filter.

To segment the slices of a model, the user selects the folder so that the images contained are listed using the function *list* and sorted in natural order with the function *sort_nat*.

Then for each slice, the user chooses iteratively, through a question dialog box, whether to segment some object. If the user selects 'yes', he can segment a new object using the function *imfreehand*. While, if he selects 'no', the data about the current slice are saved and the program passes to the next image.

With the function *imfreehand*, the coordinates x and y of the drawn path are obtained, so in order to create the binary mask with the same sizes of the image, the function *poly2mask* is used. For each slice, the binary matrix of the mask, its figure and the original image with the segmented boundary in red, are saved.

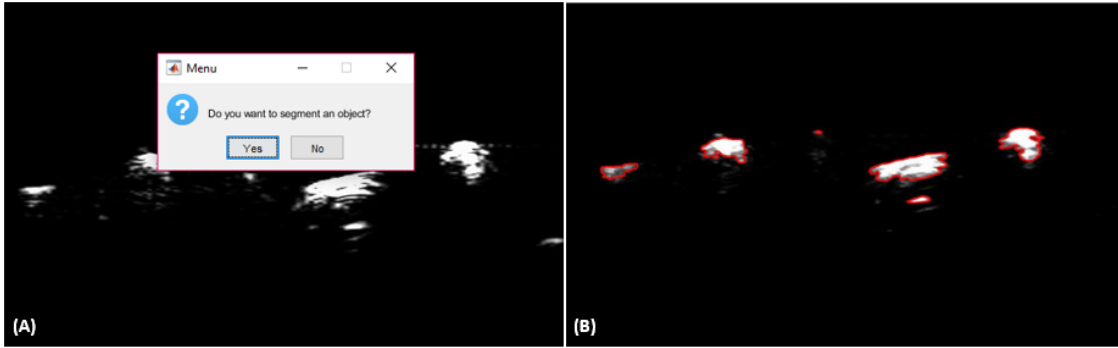


Figure 5.8: (A) Dialog box for the manual segmentation (B) Segmented image with red boundaries.

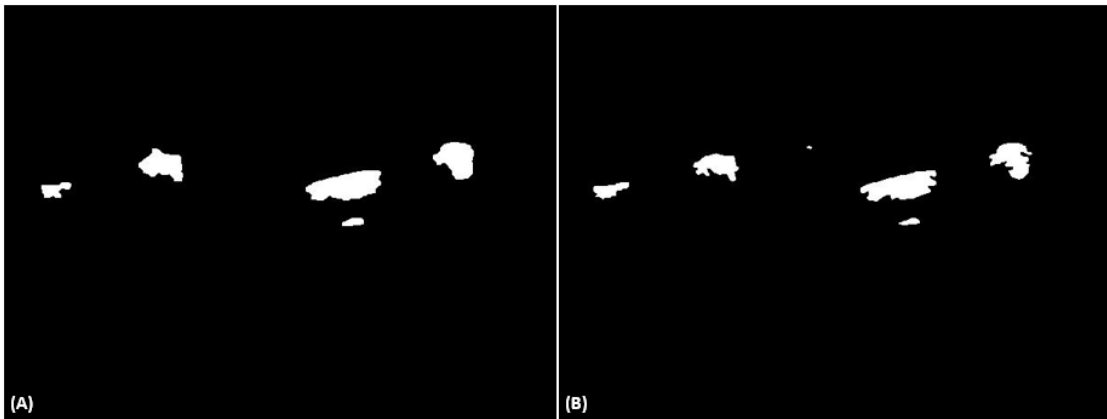


Figure 5.9: Comparison among automatic (A) and manual (B) segmentation.

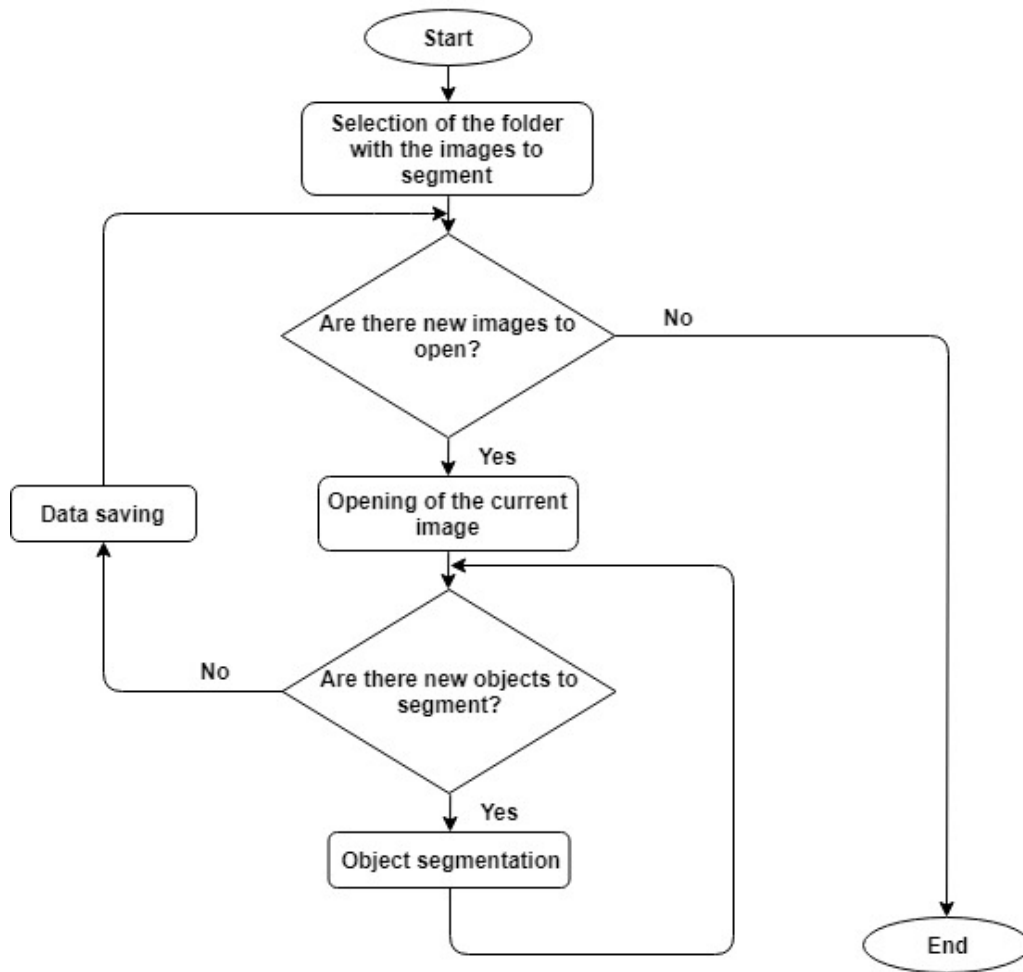


Figure 5.10: Manual Segmentation Flowchart.

5.2.3 Validation

For evaluating the performances of the automatic segmentation, statistical measures were calculated using the manual segmentation as ground truth.

Let define:

- True Positive (TP): ‘*hit*’, positive cases correctly identified. In this case the pixels correctly classified by the automatic algorithm belonging to the foreground.
- False Positive (FP): ‘*false alarm*’, negative cases identified as positive. In this case the pixels that the automatic algorithm classifies as foreground while the manual user as background.
- True Negative (TN): ‘*correct rejection*’, negative cases correctly identified. In this case the pixels correctly classified by the automatic algorithm belonging to the background.
- False Negative (FN): ‘*miss*’, positive cases identified as negative. In this case the pixels that the automatic algorithm classifies as background while the manual user as foreground [12]

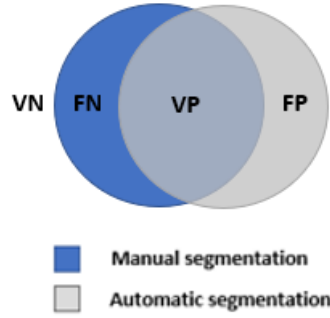


Figure 5.11: Graphic definition of VP, VN, FP and FN.

So the followed parameters are calculated:

$$Sensitivity = \frac{TP}{TP + FN} \quad (5.1)$$

Capacity of the algorithm to segment correctly pixels of foreground.

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

This parameter indicates how “useful” the results are.[12]

$$Jaccardindex = \frac{TP}{TP + FP + FN} \quad (5.3)$$

It is the ratio between the intersection of the two masks and their union.
The value of each parameter was calculated for each slice of each model and then for each model

the mean and the standard deviation were computed.

	<i>Sensitivity</i>	<i>Precision</i>	<i>Jaccard Index</i>
<i>Mean</i>	0.8450	0.8093	0.6851
<i>Standard Deviation</i>	0.1361	0.2085	0.1758

Table 5.1: Validation segmentation M_{simple} .

	<i>Sensitivity</i>	<i>Precision</i>	<i>Jaccard Index</i>
<i>Mean</i>	0.9421	0.7191	0.6783
<i>Standard Deviation</i>	0.112	0.1846	0.1781

Table 5.2: Validation segmentation $M_{complex}$.

	<i>Sensitivity</i>	<i>Precision</i>	<i>Jaccard Index</i>
<i>Mean</i>	0.7788	0.8276	0.6477
<i>Standard Deviation</i>	0.224	0.0908	0.1682

Table 5.3: Validation segmentation M_{deep} .

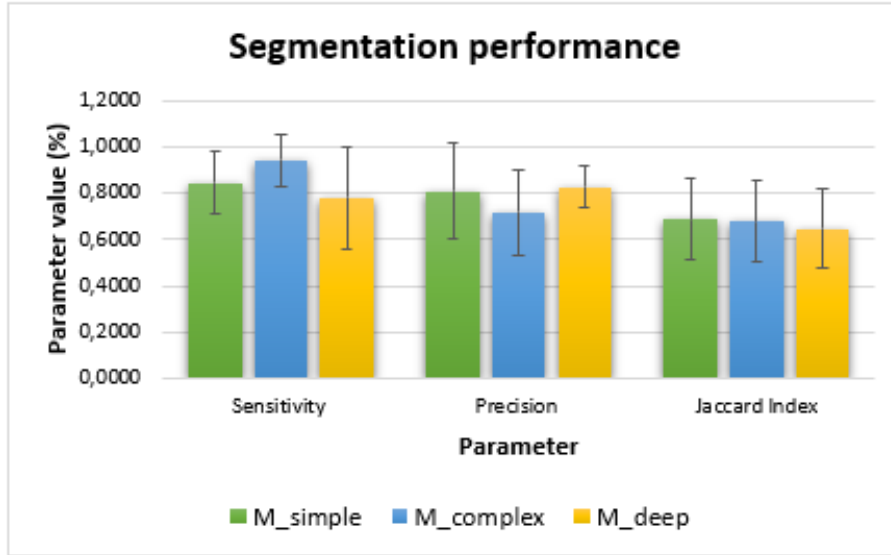


Figure 5.12: Segmentation performance evaluation for the three models: M_{simple} in green, $M_{complex}$ in blue, M_{deep} in yellow.

As it is deduced from the tables, sensitivity and precision have good values, in fact the worst value of sensitivity is around 77% for M_{deep} and the worst value of precision is 71% for $M_{complex}$. The values of the Jaccard Index are around 65%. Some differences among manual and automatic segmentation are due to the elimination of the smallest objects that isn't present in the case of the manual user. Moreover, while the user is very precise in the definition of the boundary, the algorithm because of the morphological operators is a little rough. Overall, the automatic segmentation has good performance.

5.3 Models Conversion from STL to MATLAB

To obtain the ground truth and make the validation of the algorithm implemented for vascular complexity calculation, STL models obtained with Solidworks, were converted in MATLAB matrixes.

5.3.1 Standard Tessellation Language

Standard Tessellation Language, usually abbreviated as STL, is a file format widely used in prototyping like for example in 3D printing. In this type of format, only a triangular mesh of the surface is stored in an ascii or binary file, while other information like colour, texture or material there aren't.[14]

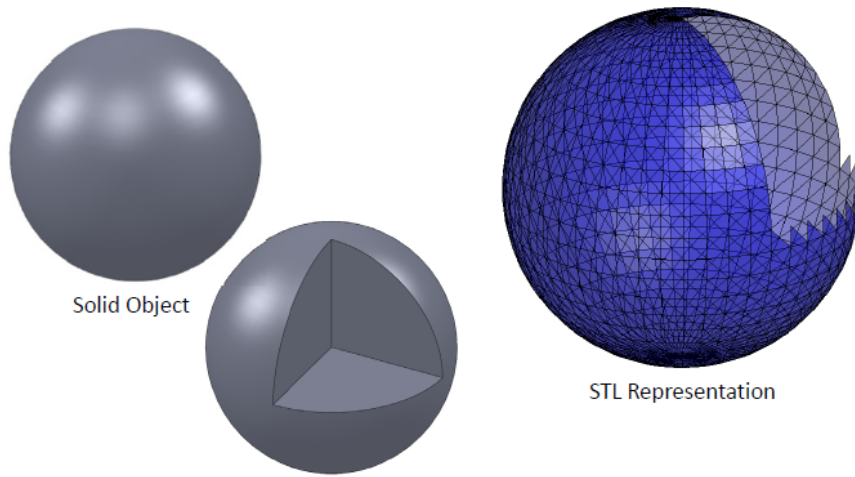


Figure 5.13: Example of an STL representation of a spherical solid object.[14]

So, inside an STL file there are all the triangles that constitute the object surface, each defined by its normal and the coordinates of its three vertices.[14]

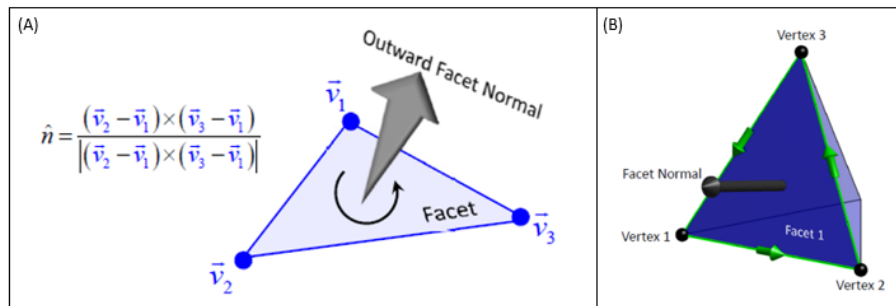


Figure 5.14: (A) Facet normal definition from vertices (B) Example of an STL structure.[14]

The normals must point towards the outside and there must be no intersections among different triangles. If there are N facets, the vertices are $\leq 3N$. The reason is that many vertices are shared by two facets, so the redundant vertices can be inserted or not. Therefore, two arrays are present: $F(N,3)$ with the array indices of the vertices that define each of the N facets and $V(\leq 3N,3)$ with the coordinates of each of the M vertices.[14]

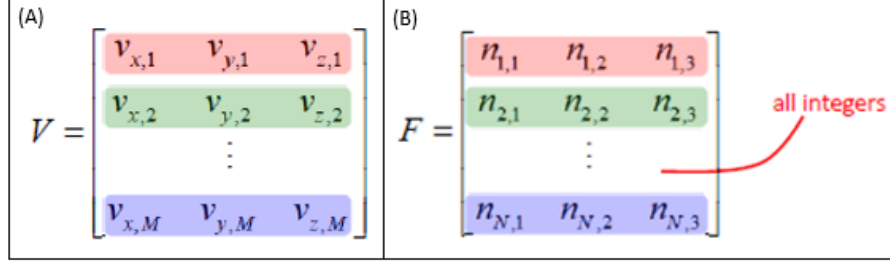


Figure 5.15: (A) Vertices matrix (B) Normals matrix.[14]

5.3.2 Function VOXELISE

The conversion from STL models of the objects to voxel models, was made with the function VOXELISE by Adam H. Aitkenhead [52] which is based on a ray tracing algorithm.

Since this passage was done for comparing results with that of the acquired volumes that report the ink contained in the vessels, the models converted and used in this step, aren't the hollow ones that were printed, but their inside part.

For each direction, resolution was calculated from the largest dimension of the object:

$$Resolution = \frac{\text{Largest dimension of the object}}{\text{Matrix size}} \quad (5.4)$$

Where matrix size is an input parameter chosen by the user before the procedure of voxelization and on the basis on which, the program allocates the necessary memory for the model.[15]

In particular, in this thesis, a conversion factor of 10 was chosen for all the directions in order to not deform the objects. So, the initial dimensions of the models in mm, were multiplied by 10 and fixed to the nearest integer, seeing as the matrix sizes must be integers. In the tables, real model sizes, matrix sizes and resolution values are reported. As it's shown in some cases the resolution isn't exactly 0,1 because of approximation, but the error is very little.

Direction	Model size (mm)	Model size (pixel)	Resolution
x	6.9184	69	0.1
y	33.2243	332	0.1
z	37.7824	378	0.099

Table 5.4: Conversion Resolution M_{simple} .

<i>Direction</i>	<i>Model size (mm)</i>	<i>Model size (pixel)</i>	<i>Resolution</i>
x	10.7786	108	0.099
y	33.2243	332	0.1
z	37.7824	378	0.099

Table 5.5: Conversion Resolution $M_{complex}$.

<i>Direction</i>	<i>Model size (mm)</i>	<i>Model size (pixel)</i>	<i>Resolution</i>
x	30.1146	301	0.1
y	37.6454	376	0.1
z	39.6774	397	0.099

Table 5.6: Conversion Resolution M_{deep} .

For the voxelisation process, the first thing to do is the reading of the STL file to obtain the two arrays with normals and vertices of the facets. So, the algorithm starts understanding if the file is ascii or binary and then extracts the needed information from it. In particular, the discrimination is made because a binary file has always a dimension of $84 + (50 \cdot n)$ bytes (84 bytes for the header and 50 for vertices and normal of each facet, where n is the number of facets), while an ascii file starts always with the word ‘solid’ and finishes with the word ‘endsolid’. Then, since Solidworks doesn’t use the z-axis as the vertical axis, a reorientation of the model was made with these commands[14]:

$$\begin{aligned}
 Y &= V(:, 2) \\
 V(:, 2) &= V(:, 3) \\
 V(:, 3) &= Y
 \end{aligned}$$

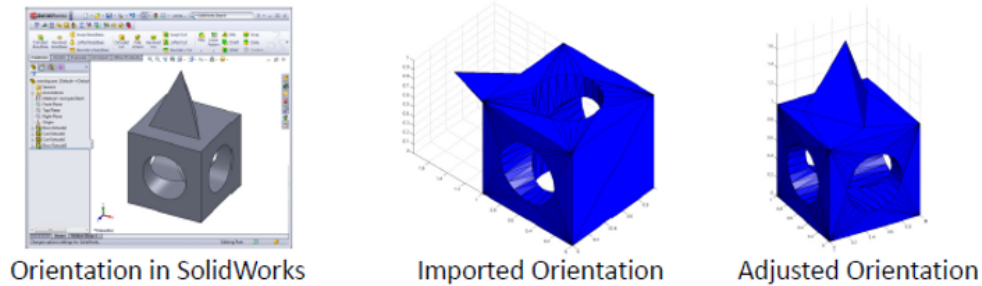


Figure 5.16: Reorientation of the model from Solidworks to MATLAB.[14]

After preparing the grid with the indicated dimensions, a set of rays is passed through the object along a specific axis to compute the intersections with the facets. These rays exactly, pass through voxels centers that are in the grid perpendicular to the chosen axis. So, it's made a loop through each pixel in order to consider a ray for each grid point from the minimum to the maximum of the two grid directions.[15][16]

For each iteration, all the possible crossed mesh facets are found. For every one of them it's verified that the ray is on the same side of the opposing vertex with respect to every edge of the facet. Instead if the ray crosses exactly on a vertex, it's verified that the near facets have normal components with the same or the opposite direction of the ray.

Thereafter the z coordinates of the points where rays cross the mesh, are individuated. First, there is the definition of the equation that describes the plane of the facet:

$$\begin{aligned}
 Ax + By + Cz + D &= 0 \\
 \text{where } A &= y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\
 B &= z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \\
 C &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\
 D &= -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1)
 \end{aligned}$$

Then the equation is solved with the coordinates of the points to determine z.

Last, all the voxels between an odd and the next intersection point are filled.

For reducing voxelization errors, it's convenient to repeat this procedure for each direction (x,y,z) and to make finally a polling for each voxel. In particular, 1 is assigned if almost two among three votes are 1. Ideally, the same result for each direction, filled or empty, is desirable.[15]

There are three different possible errors: missing thin sections, odd number of intersections for sections thinner than voxel size, and discretization errors.[16]

The first there is when a section is completely between two rays without intersections and so it is missed.[16] An odd number of intersections instead, is related to an error in the STL file or to a very thin section of one single voxel. The discretization error, that can be reduced with a higher resolution, is defined as the biggest distance between the surface of the object and the surface of the nearest voxel, so it is smaller than voxel size. If voxel size is less than half of the minimum section thickness, the possibility of thin sections disappearing is removed.[15]

Remember that even STL file is already an approximation of the object surface made with the CAD software.

At the end, the voxel model is a set of some binary (0: empty or 1: filled) voxels aligned to the axes and stored in different layers which compose the space volume in $Z^3 = \{v = (i,j,k) | (i,j,k) < (I_{max}, J_{max}, K_{max})\}$. Each voxel is represented by its centroid (i,j,k) and has a location that is characterized by a precise index in the array.[15]

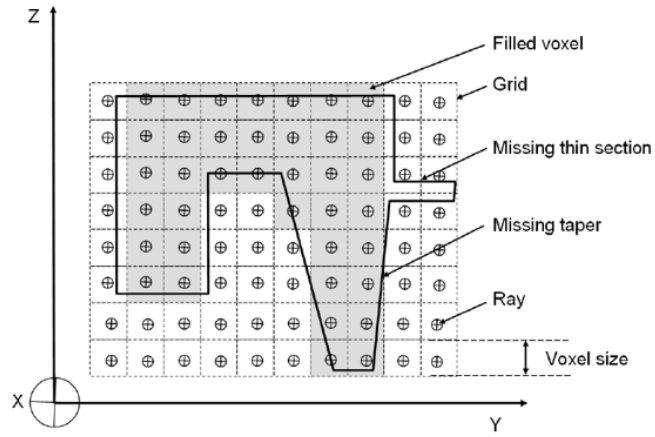


Figure 5.17: Scheme of voxelisation procedure and some possible errors.[15]

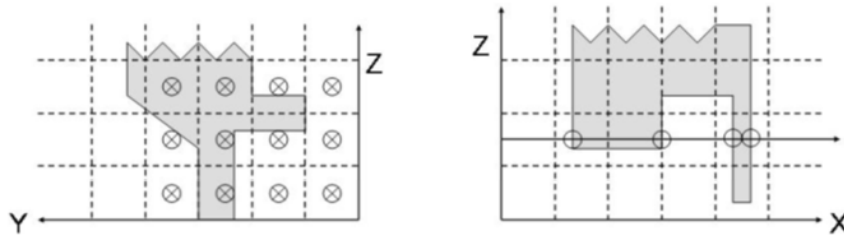


Figure 5.18: Some errors due to thin sections.[16]

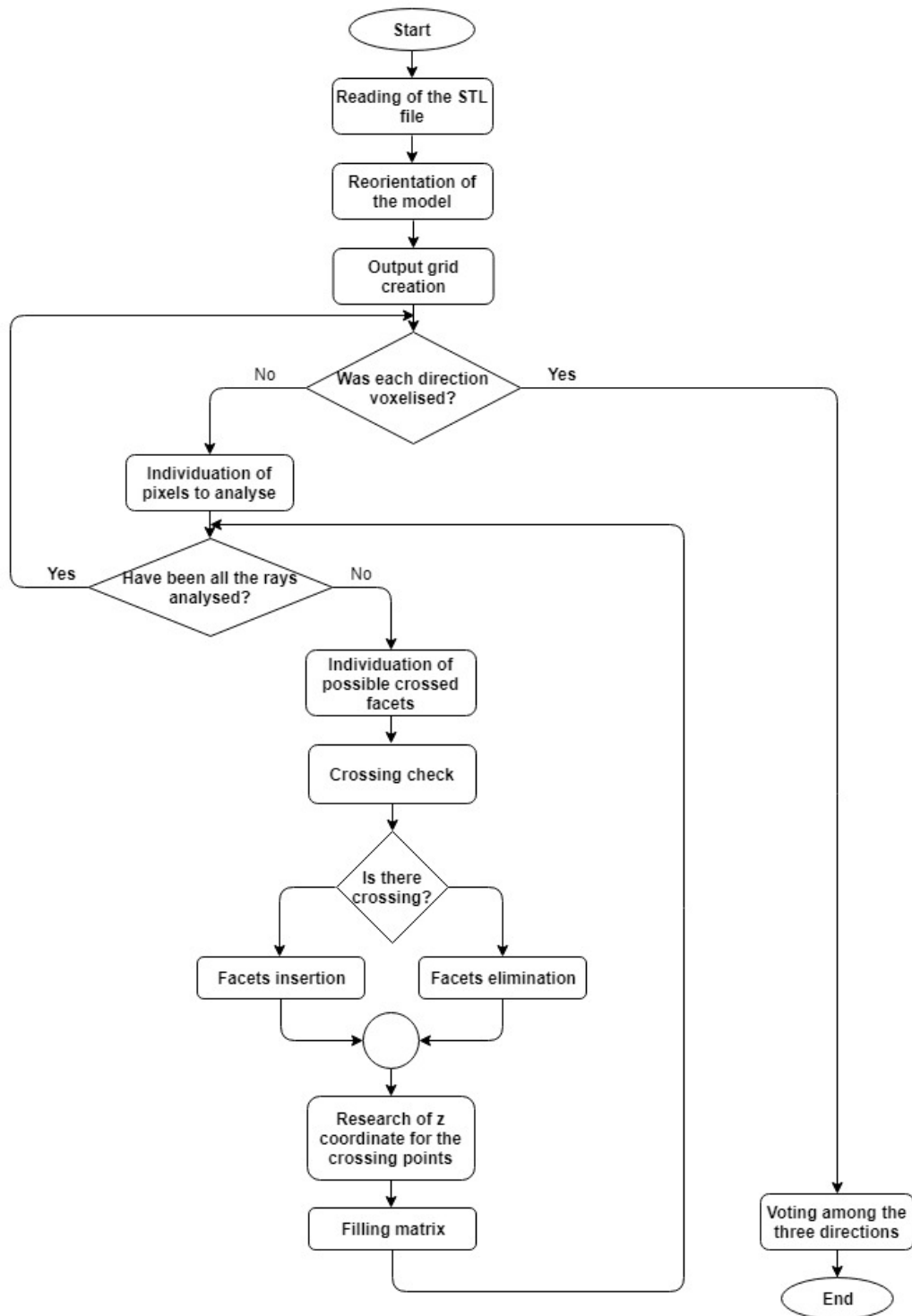


Figure 5.19: Voxelise Flowchart.

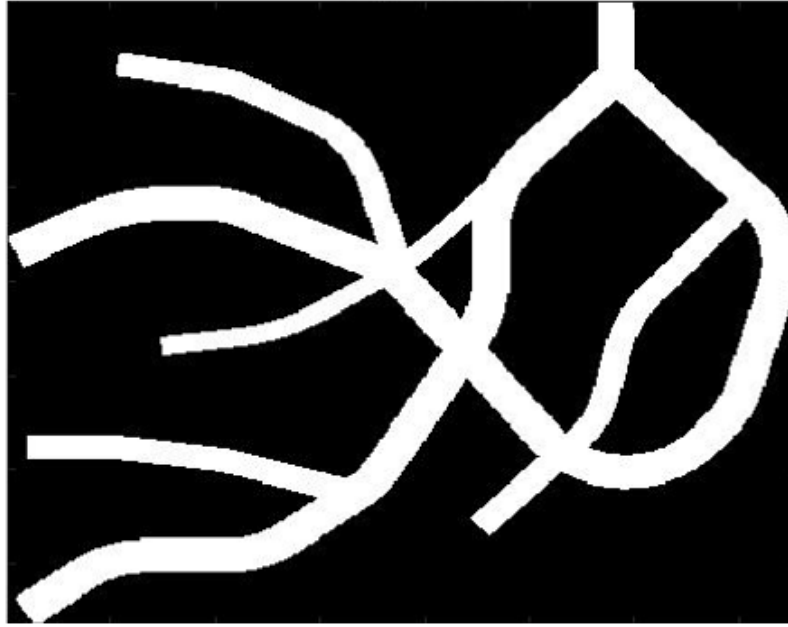


Figure 5.20: MIP of converted M_{simple} from Solidworks.



Figure 5.21: MIP of converted $M_{complex}$ from Solidworks.

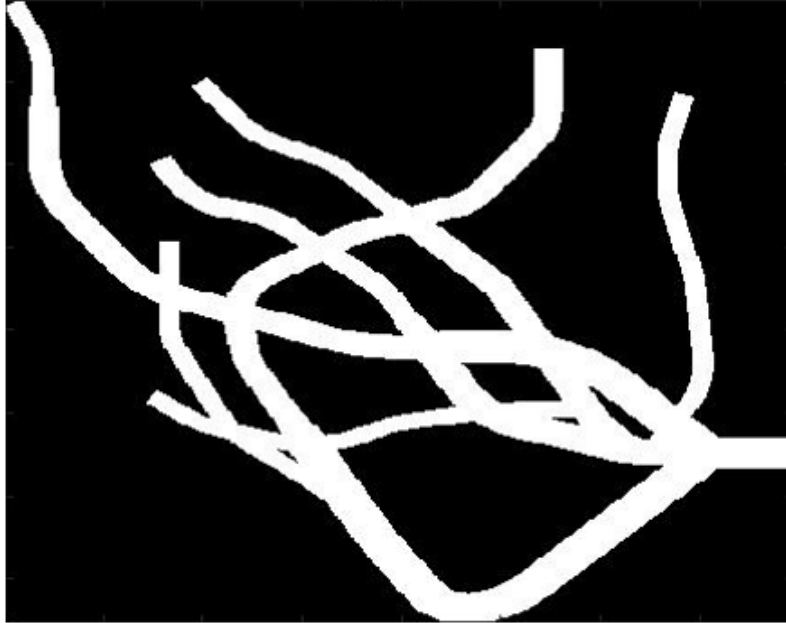


Figure 5.22: MIP of converted M_{deep} from Solidworks.

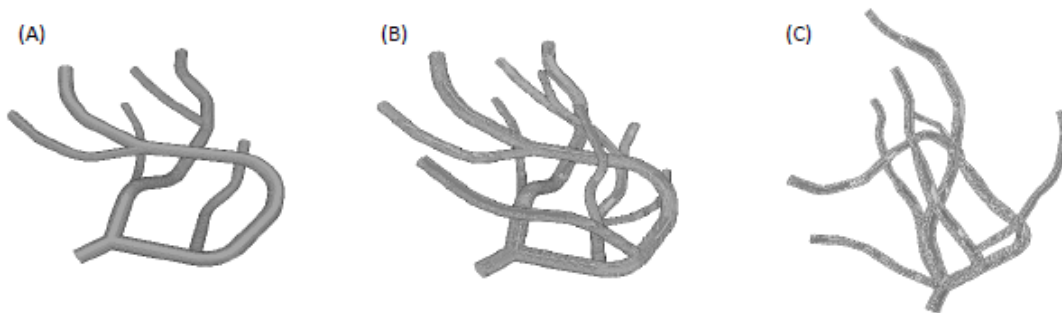


Figure 5.23: Converted models: (A) M_{simple} (B) $M_{complex}$ (C) M_{deep} .

Chapter 6

Skeletonization and Parameters Calculation

6.1 Skeletonization

6.1.1 Function bwskel

For the skeletonization, the Matlab function *bwskel* was used. It is based on the medial axis extraction algorithm by *Lee et al* [17] that is implemented preserving the topology and the Euler number. To understand the principal aspects of this algorithm, it's necessary to introduce some basic definitions. The geometry is represented with an object stored in a 3D array of sizes k_{max} , j_{max} and i_{max} where each volume element v is represented with its centroid (k, j, i) . The value of v can be 0 or 1. S is the subset of all the points with value 1, while its complement, so the subset of all the points with value 0 is \bar{S} . A border point is a point v in S that has at least one of its 26-neighbors in \bar{S} . The 26-connected neighbourhood is defined as $N_{26}(v = (k, j, i)) = \{(w, v, u) | w - k| + |v - j| + |u - i| = 1\}$, while a 26 path with length n is a points sequence where v_i is 26 adjacent to v_{i-1} for each i . [17]

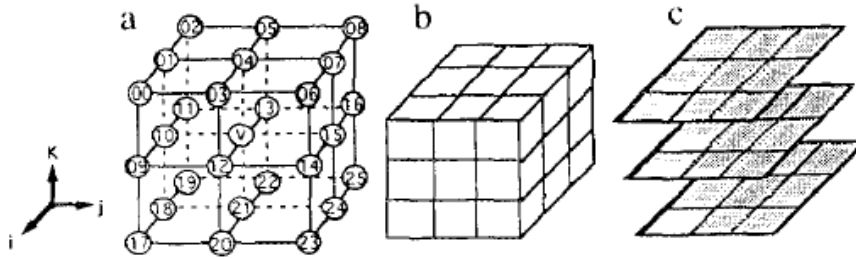


Figure 6.1: 26-neighbourhood for the point v : indices representation (a) and other two representations (b) and (c). [17]

The thinning operation can be performed with kernels, applying structural elements to the image, or with a decision tree, that if well designed is faster. The solution used in this function is based on a decision tree which handles all the 3D possible combinations in a 26-neighbourhood

that are $2^{26} = 67.108.864$ (Figure 6.2). In this way, for every iteration, it can identify all the detectable surface points. This octree representation is called adjacency tree of $N_{26}(v)$ and it performs recursive division of the cubic volume into eight sub-cubes called octants. In the tree, the root node is v , that is the center voxel of $N_{26}(v)$, while the leaves are data points. It is used to calculate the number of $N_{26}(v)$ connected objects in an efficient manner, because through the connection of the overlapped points in the octants, adjacency relations are individuated.[17]

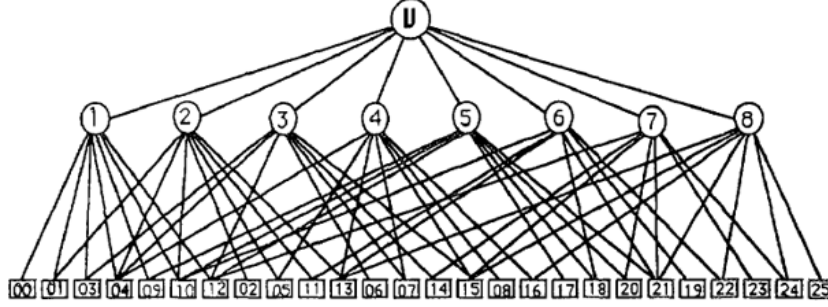


Figure 6.2: Tree representation of the 26-neighbourhood $N(v)$. [17]

Voxels are removed iteratively from the surface of the object only if removing them, some defined conditions are satisfied. In particular:

- The point has to be a surface voxel. In particular, the algorithm is divided in six sub-cycles depending on the type of border point in order to have symmetry. A border point $v = (k, j, i)$ is defined N(north), S(south), W(west), E(east), U(up) or B(bottom) if the cubic index 13, 12, 10, 15, 4 or 21 is zero respectively.
- The point doesn't have to be an endpoint because the centerline has to remain. A point is considered an endpoint if it is connected to just one other voxel.
- The Euler number doesn't have to change, so holes aren't created.
The global formula of the 3D Euler characteristic is $\chi(S) = O(S) - H(S) + C(S)$
With $O(S)$: number of connected objects, $H(S)$: number of holes, $C(S)$: number of cavities.
Euler number is computed using a lookup table.
- The voxel has to be a simple point, so the connectivity doesn't change.
A simple point is a border point whose removal doesn't change the number of connected objects and holes for S and \bar{S} .

So, for the deletion, only simple border points are considered in order to preserve the topological properties. However, there are many situations in which removing simultaneously these points, the original object is completely eliminated. To solve this issue, a sequential re-checking procedure is performed: Let's consider R as the simple border points and $Q = S - R$, therefore the remaining points removing R from S . Each point $v \in R$ is re-examined. If the path connectivity in $R \cap N(v)$ and $Q \cap N(v)$ is preserved, v is deleted. [17] [53]

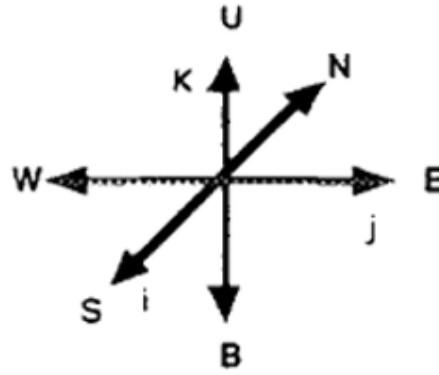


Figure 6.3: The six directions of the sub-cycles.[17]

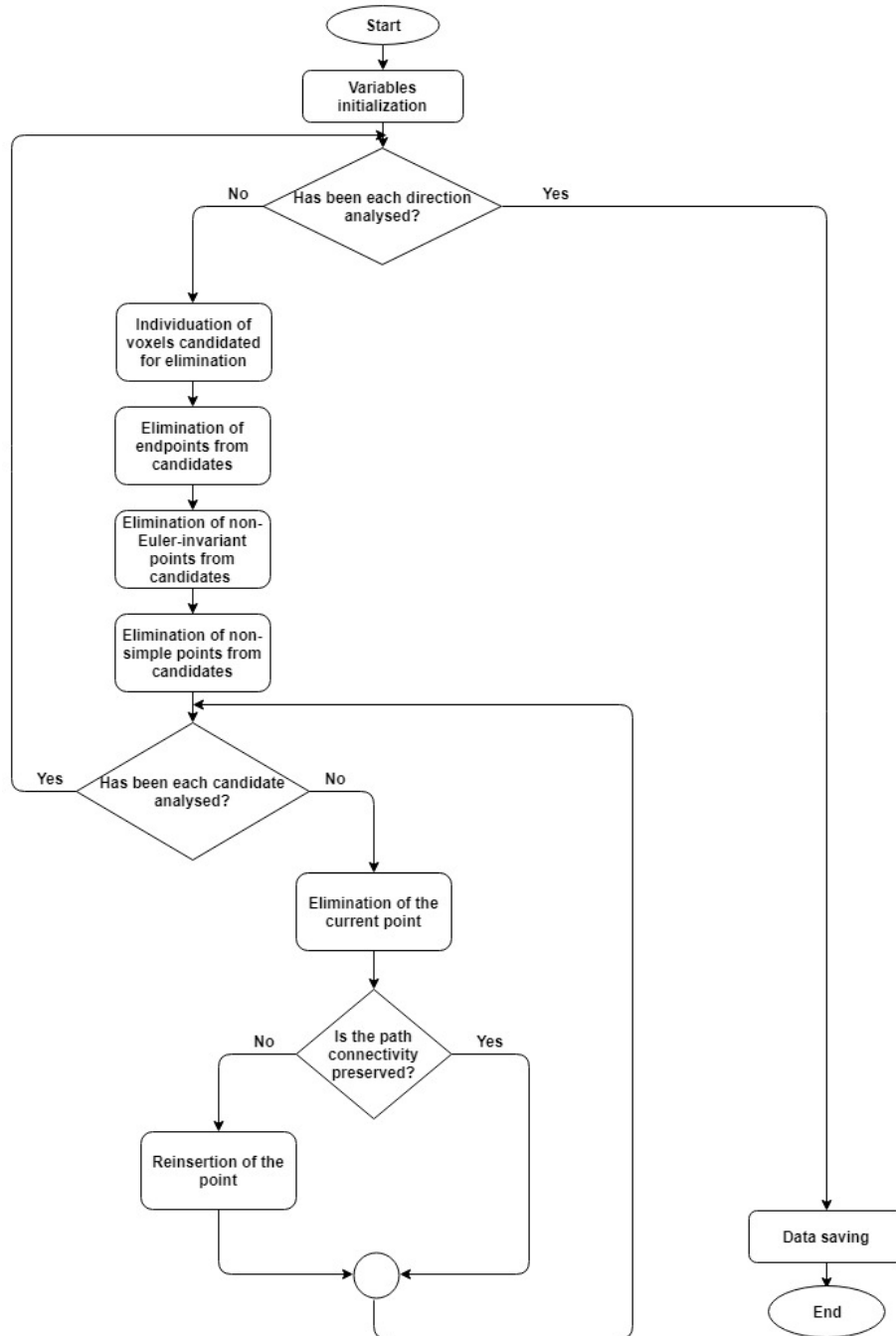


Figure 6.4: Skeletonization flowchart.

6.1.2 Skeleton cleaning

An algorithm was implemented with the aim to correct the defects of the skeletonization and to refine the structure removing the smallest branches.

In some areas there are accumulations of points. For removing them, the branchpoints are identified and when, among them, there are connected objects with a value bigger than 10 pixels, they are removed. To remove the remaining voxels of the accumulation boundaries, the function *bwareaopen* with threshold 5 is used. It deletes all the objects that have fewer pixels.

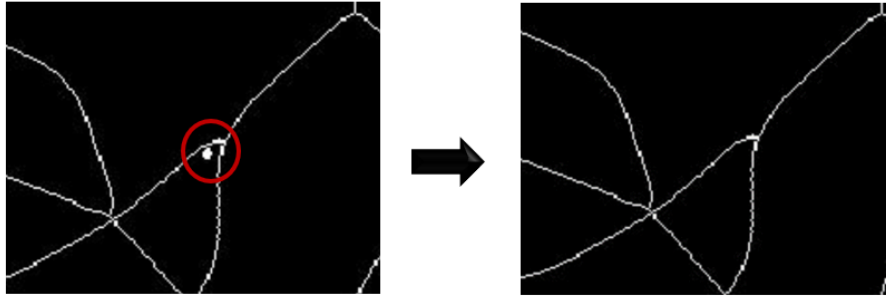


Figure 6.5: Points agglomeration in the skeleton on the left and its elimination on the right.

Thereafter the branches with length smaller than a defined threshold are removed. There is a for loop for the endpoints of the skeleton, for each of them, the distance from all the branchpoints is measured with the function *bwdistgeodesic* and the minimum among them is searched. If this distance is smaller than the threshold, all the voxels of the path from the branchpoint to the endpoint are removed except the branchpoint itself. In the case where the branchpoint of a branch isn't individuated but the endpoint yes, as in the case where the branch is a unique pixel, a big piece of skeleton could be deleted wrongly. So, for each removed branch, there is a control about the number of connected objects, if the number changes respect to the step before the pixels are restored to 1 toward the number of connected objects returns the initial one.

The skeletons of the original models don't contain many defects, so a threshold of 20 is used, instead for the models obtained from the automatic and the manual segmentations two thresholds are used: 20 and 40. Even if there are other branches that should be removed it's impossible to increase the value for not removing important parts of the structure.

Also in this case, ImageJ was used to create videos of the original and the cleaned skeletons.

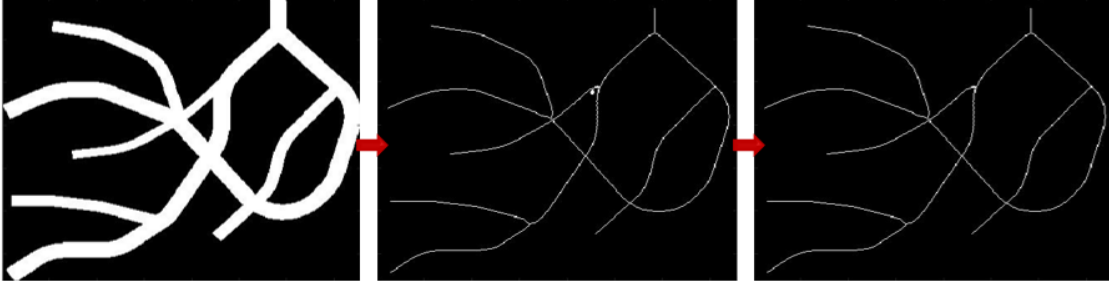


Figure 6.6: M_{simple} - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.

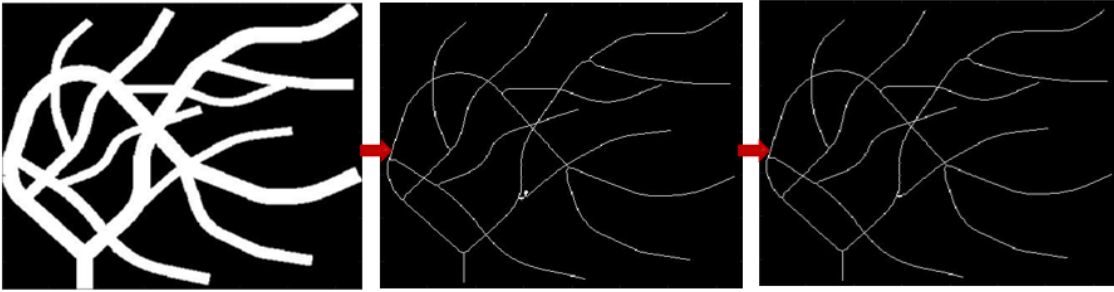


Figure 6.7: $M_{complex}$ - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.



Figure 6.8: M_{deep} - Original model MIP on the left, original skeleton MIP in the centre, final skeleton MIP on the right.

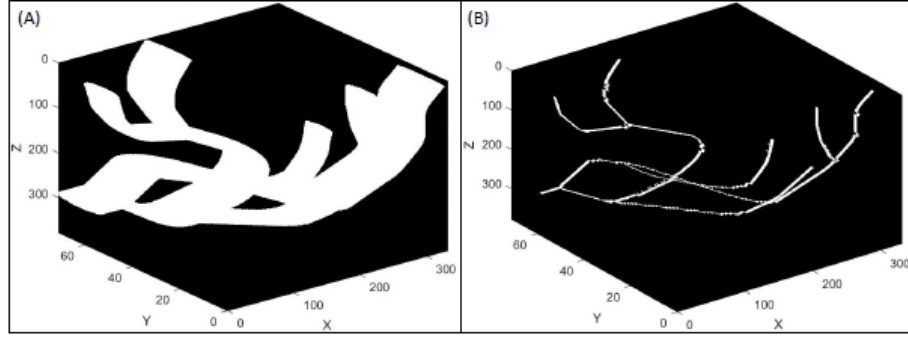


Figure 6.9: M_{simple} : (A) Original model and (B) corresponding skeleton.

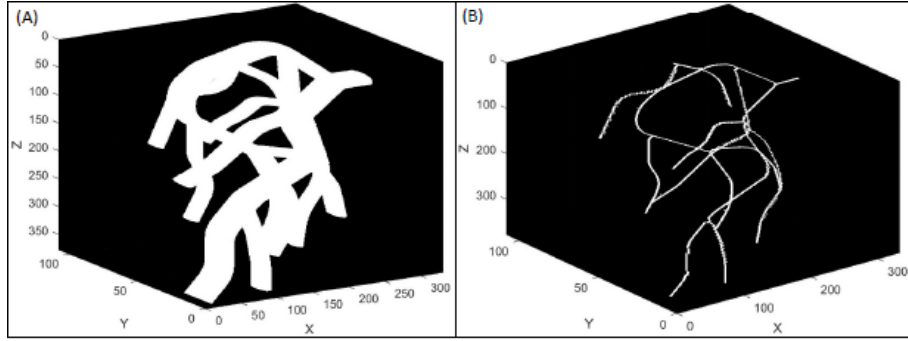


Figure 6.10: $M_{complex}$: (A) Original model and (B) corresponding skeleton.

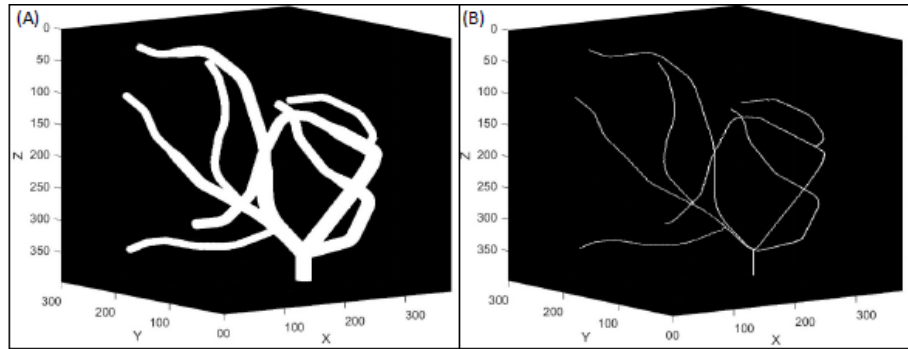


Figure 6.11: M_{deep} : (A) Original model and (B) corresponding skeleton.

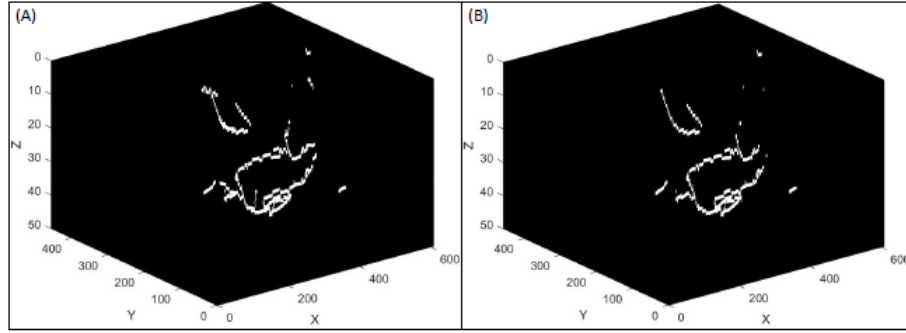


Figure 6.12: M_{simple} : (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.

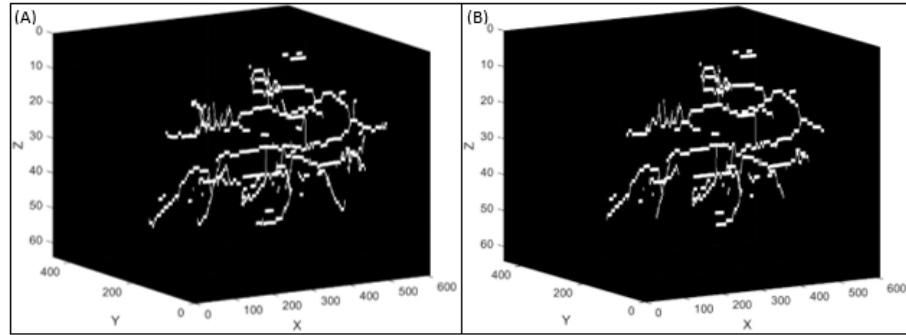


Figure 6.13: $M_{complex}$: (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.

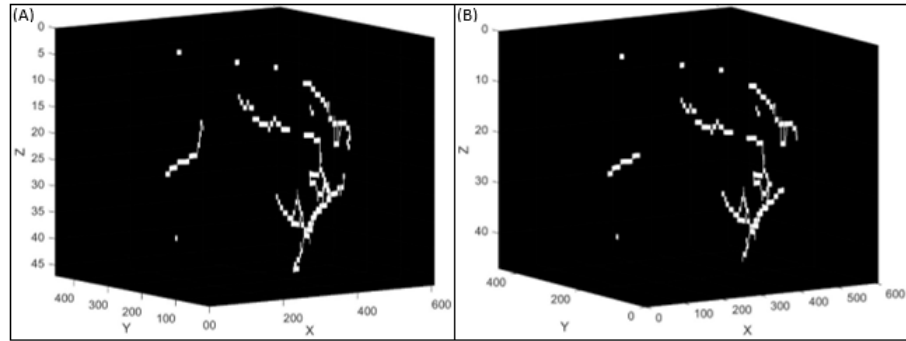


Figure 6.14: M_{deep} : (A) Automatic skeleton with threshold 20 (B) Automatic skeleton with threshold 40.

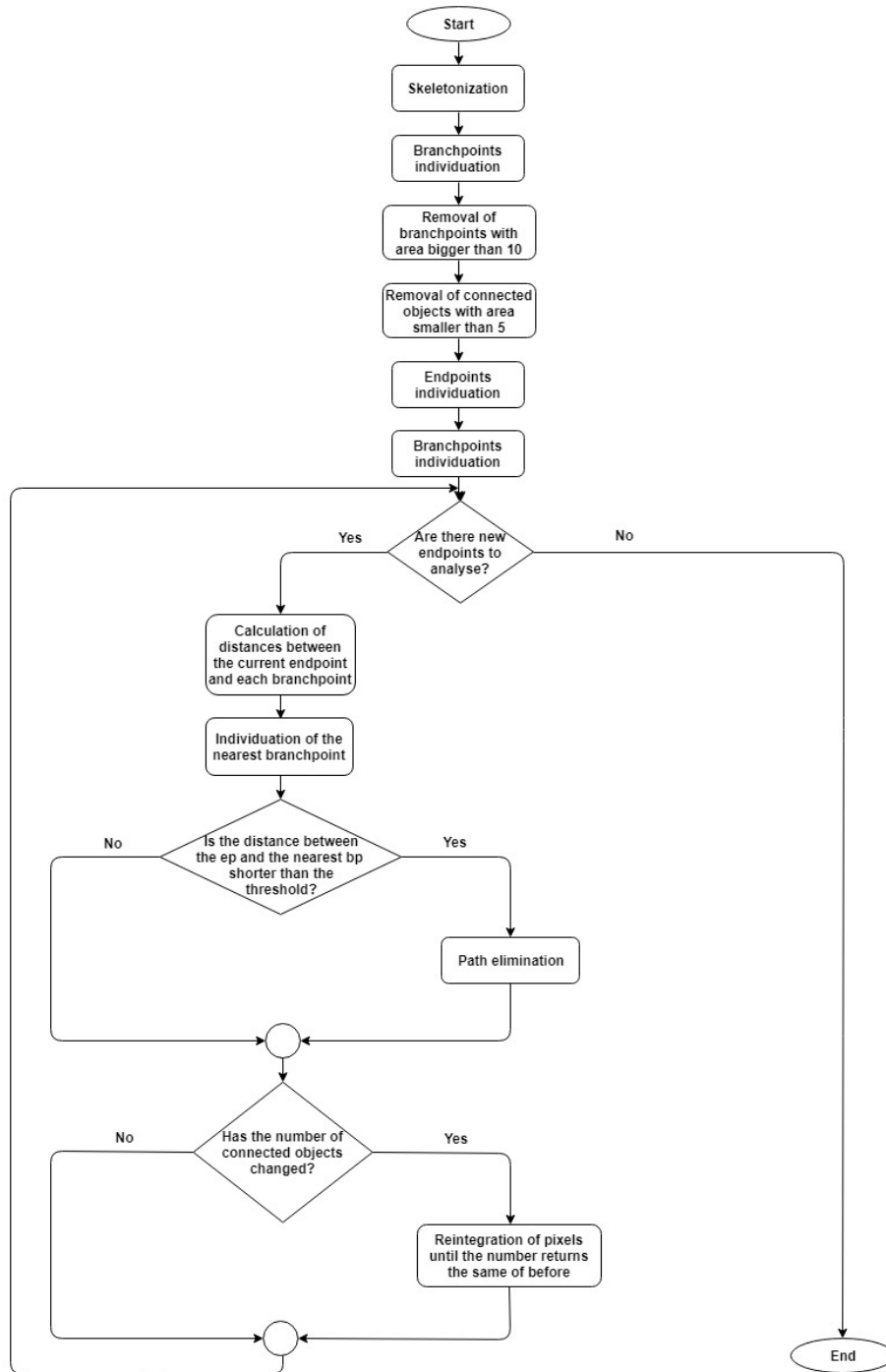


Figure 6.15: Skeleton Cleaning Flowchart.

6.1.3 Branchpoints search

The Matlab function *bwmorph3* bases its branchpoints search on the definition for which a voxel is considered a bp if it has at least three neighbours in its 26-connected set. However, there are cases where having three neighbours isn't sufficient, so it was necessary to modify this function.

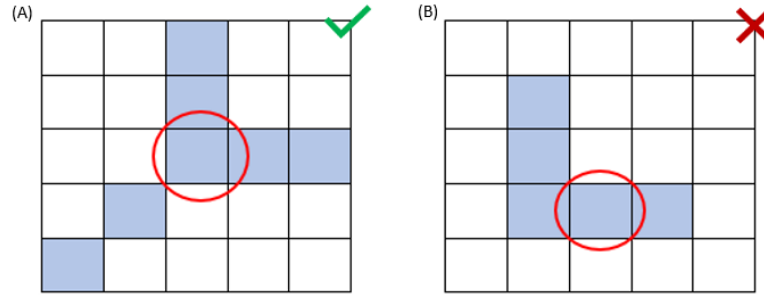


Figure 6.16: (A) Example where the point with three neighbours is a branchpoint (B) Example where the point with three neighbours isn't a branchpoint.

In the algorithm implemented, there is a for loop that takes into account each branchpoint found by the function and when the voxel (coloured in blue) has three neighbours, some controls are made in order to decide if it is really a branchpoint or not.

For each of these three points (coloured in yellow), the 26-neighbourhood is calculated with the purpose to establish if there are three separated branches. The three neighbourhood are put in a zeros matrix with the sizes of the skeleton, removing voxel indagated (blue), its three neighbours (yellow) and the duplicates (Figure 6.17). In this way, only red points remain and it is checked that among them, everyone isn't in the neighbourhood of the others, so that there are three different connected components. Otherwise the blue point is deleted from the branchpoints. At the end of the loop, the remaining points are dilated with a spherical structuring element with radius 2 and so the connected components are counted to define the correct number of bp.

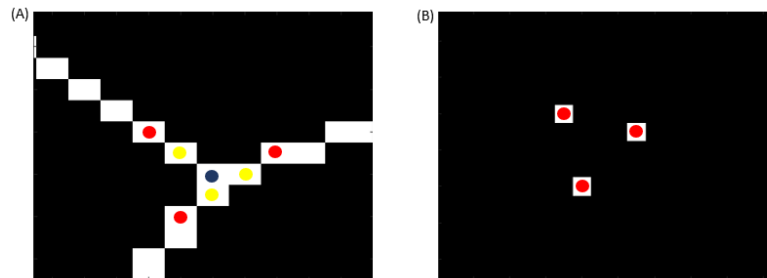


Figure 6.17: (A) Example of a branchpoint with three branches, the branchpoint is the blue voxel, its neighbours are the yellow, while the red are the neighbours of the neighbours (B) Matrix with the neighbours to control if there are three separated connected objects.

Through this algorithm, the correct branchpoints of the three original models are found.

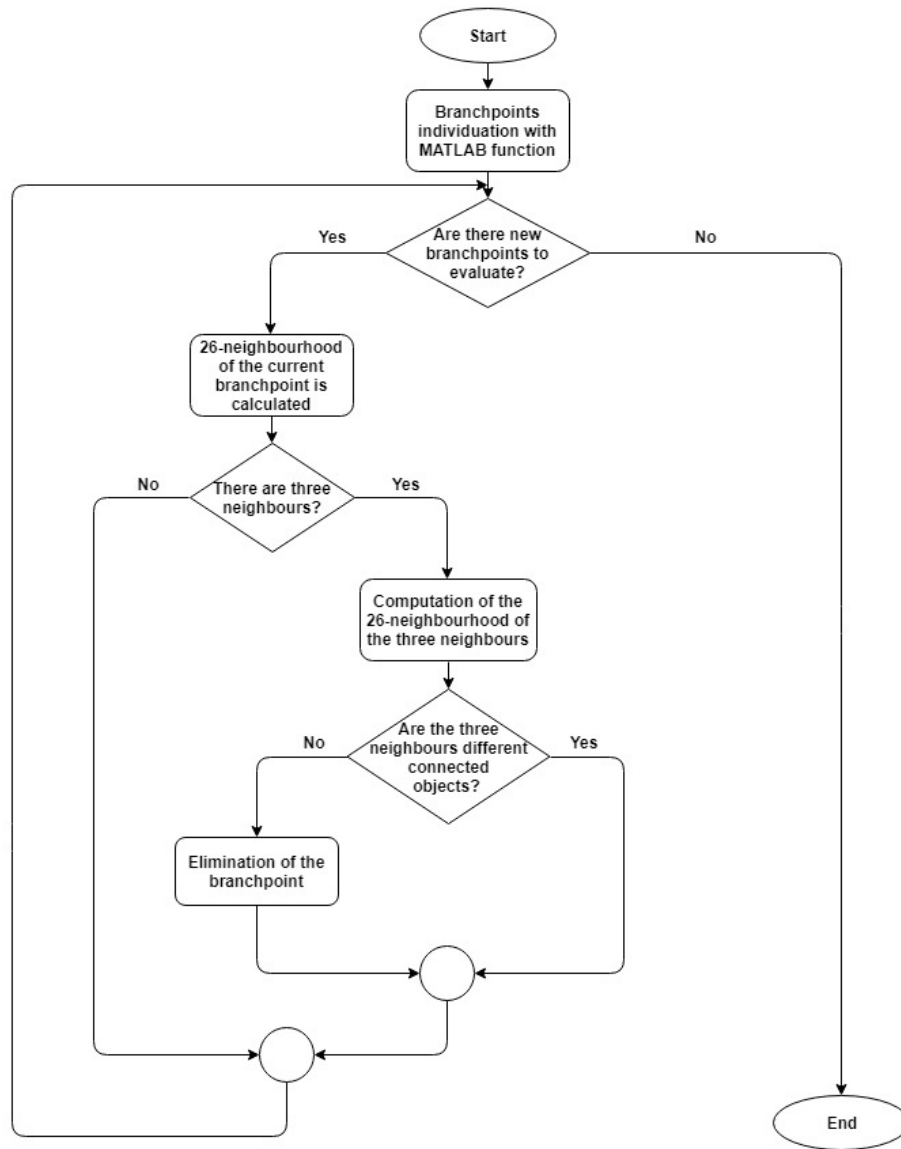


Figure 6.18: Branchpoints search flowchart.

6.1.4 Endpoints search

Also for endpoints, the Matlab function *bwmorph3* is based on a definition that in practice doesn't work well, for which a voxel is considered an ep if it has maximum one neighbour in its 26-connected set. In particular, not all the endpoints are found, because they could have more than one neighbour. So, even in this case, a modification of the function was made.

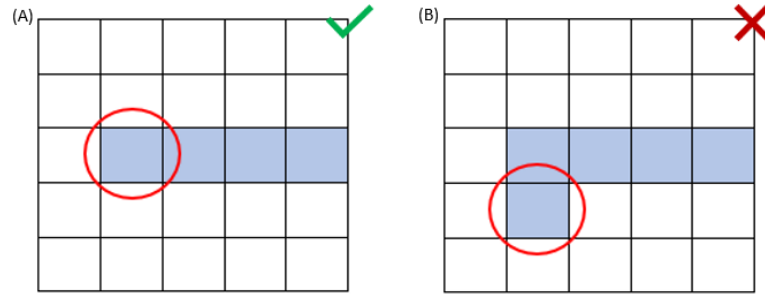


Figure 6.19: (A) Example where the endpoint is correctly individuated (B) Example where the endpoint is missed because it has two neighbours.

In the algorithm implemented, there is a for loop that considers each point of the skeleton. When the point has two neighbours in its 26-connected set, the latter is multiplied by a 3 by 3 matrix (figure 6.20) in order to associate to them an index depending on their position respect to the indagated point. Thereafter, if the two neighbours have the same index the voxel isn't an endpoint, if the indexes are a "1" and a "3" the voxel is an endpoint if the neighbours have the same third coordinate, if the indexes are a "1" and a "2" or a "2" and a "3" the voxel is an endpoint if the neighbours have the same first or second coordinate and they are near.

(A) Before slice	(B) Pixel slice	(C) After slice																											
<table><tr><td>3</td><td>2</td><td>3</td></tr><tr><td>2</td><td>1</td><td>2</td></tr><tr><td>3</td><td>2</td><td>3</td></tr></table>	3	2	3	2	1	2	3	2	3	<table><tr><td>2</td><td>1</td><td>2</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>2</td><td>1</td><td>2</td></tr></table>	2	1	2	1	0	1	2	1	2	<table><tr><td>3</td><td>2</td><td>3</td></tr><tr><td>2</td><td>1</td><td>2</td></tr><tr><td>3</td><td>2</td><td>3</td></tr></table>	3	2	3	2	1	2	3	2	3
3	2	3																											
2	1	2																											
3	2	3																											
2	1	2																											
1	0	1																											
2	1	2																											
3	2	3																											
2	1	2																											
3	2	3																											

Figure 6.20: Matrix indices for the neighbourhood and the determination of the endpoints - (A) Before slice (B) Pixel slice (C) After slice.

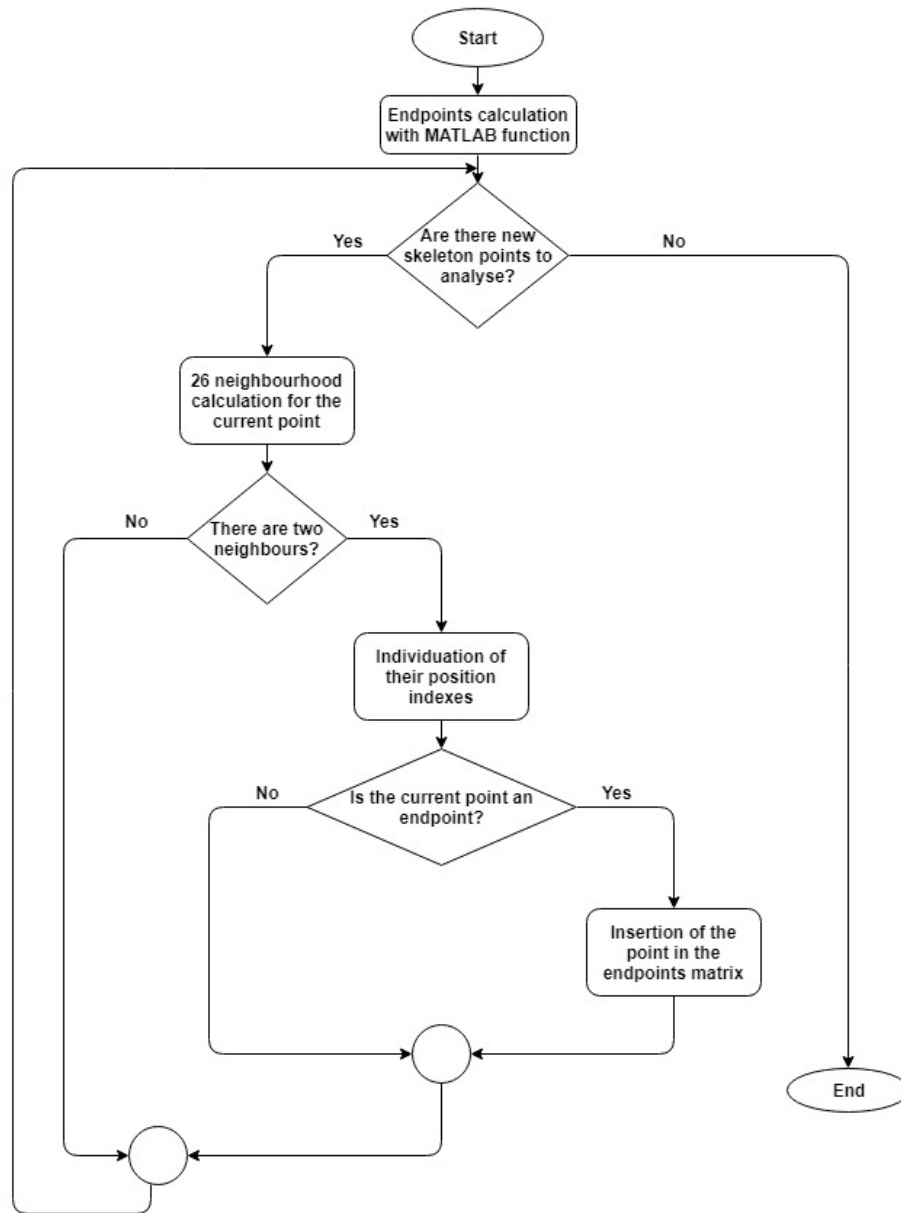


Figure 6.21: Endpoints search flowchart.

In the table are reported the number of branchpoints and endpoints for the three models:

	<i>Branchpoints</i>	<i>Endpoints</i>
M_{simple}	5	7
$M_{complex}$	9	11
M_{deep}	4	8

Table 6.1: Number of branchpoints and endpoints of the three models.

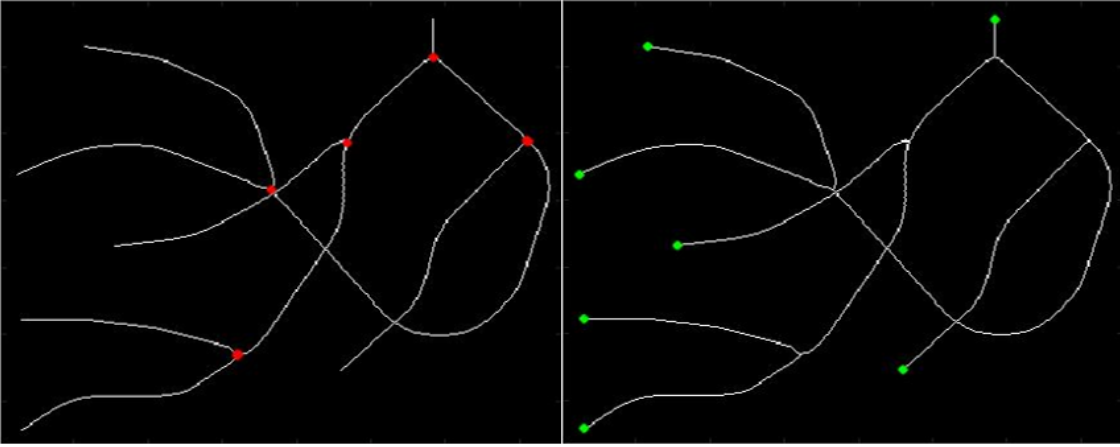


Figure 6.22: M_{simple} - Branchpoints in red and Endpoints in green.

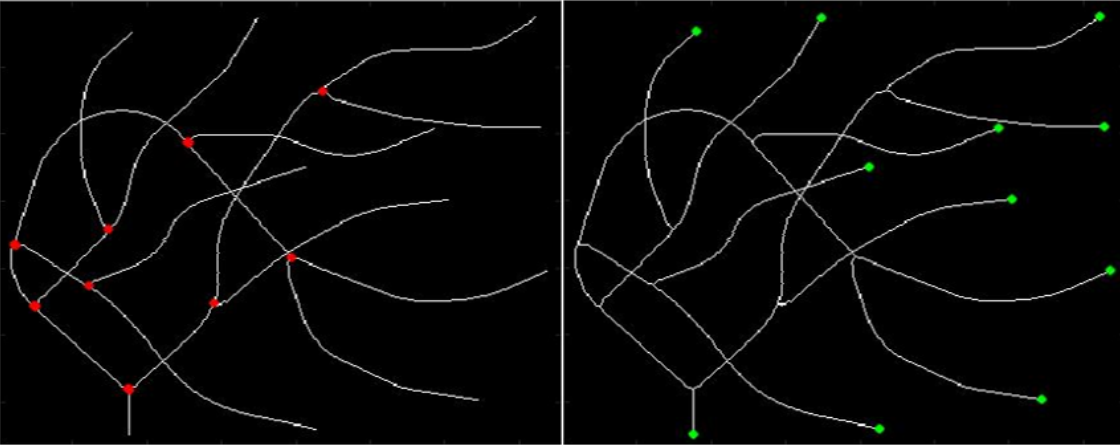


Figure 6.23: $M_{complex}$ - Branchpoints in red and Endpoints in green.

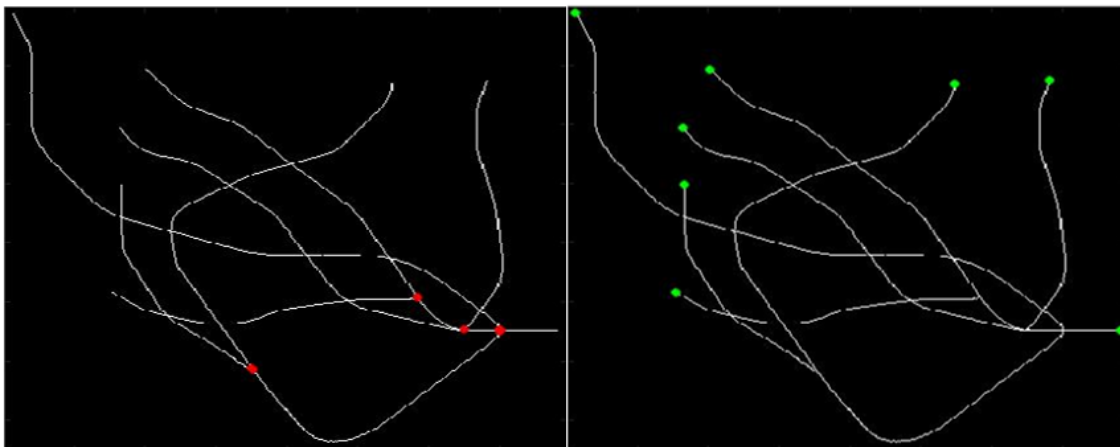


Figure 6.24: M_{deep} - Branchpoints in red and Endpoints in green.

6.2 Breadth First Search

For individuating trees and branches of the models, the Breadth First Search was used. The Breadth First Search is an efficient and simple algorithm that exploits the breadth-first graph traversal to process digital images. Many examples of BFS applications can be found in literature, such as morphological operations, skeletonization and distance-transform calculation. In each of them, image regions are investigated employing a queue for storing the unprocessed pixels and removing elements gradually when they are analysed. This allows to use memory in an efficient way. Actually, the image remains a matrix, but the graph is useful for understanding the search procedure.[54]

Let define an image region as a group of pixels with the same colour, where a continuous path of neighbours exists for every couple of pixels. It's possible to imagine the matrix as a graph, where nodes are pixels and links represent neighbourhoods among them (Figure 6.25). So, instead by doing a raster scan, pixels can be investigated in a breadth-first manner. The algorithm starts from a selected node and then automatically searches the others as long as all the nodes have been analysed. Moreover, pixels storage in a queue, leads to the consequence that this search is suitable for the problems where it's necessary to operate layer by layer. Obviously, it's required to know the position of neighbours of each element in the matrix.[54]

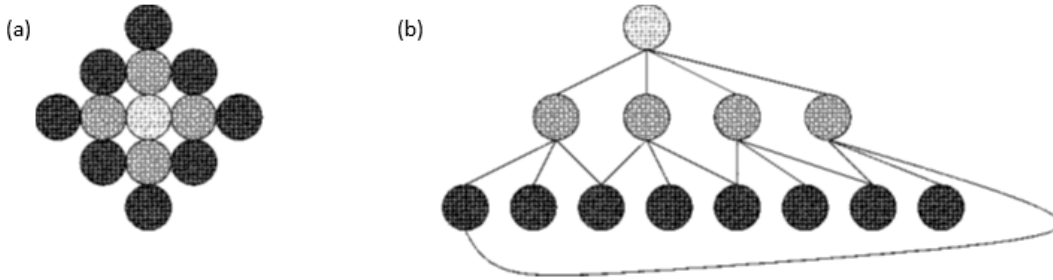


Figure 6.25: Raster image (a) seen as a graph (b).

A customized BFS implementation [23] was used in order to map the skeleton as a set of vascular trees on which were evaluated tortuosity and morphological parameters.[23]

Starting from the skeleton of the specific model, the connected objects are individuated and that with area smaller than a threshold are removed. Two thresholds were used in this thesis: 20 and 50. Then for each of the remaining objects the branches are found. First the longest path is searched using the function *bwdistgeodesic* among each endpoint and each skeleton element. The pixels of this path are deleted from the skeleton and put in another matrix with a value of 1. After, the connected objects are again searched in the remaining skeleton and all the elements with area smaller than 10 are removed. With a for loop every remaining endpoint is considered, for each of them the longest path is found, removed from skeleton and put in the other matrix with an index that increases at each iteration. After that it's verified that there aren't paths without endpoints, otherwise the new endpoints are calculated and it is made the same thing of before.

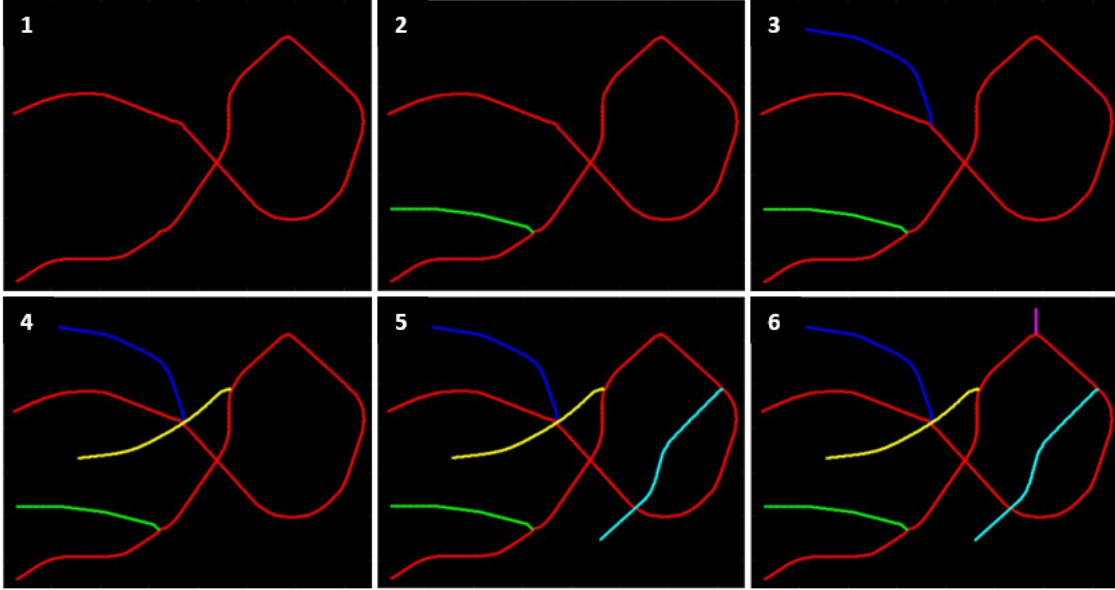


Figure 6.26: BFS algorithm on the M_{simple} - In order, the individuation of the six branches (red, green, blue, yellow, cyan and magenta).

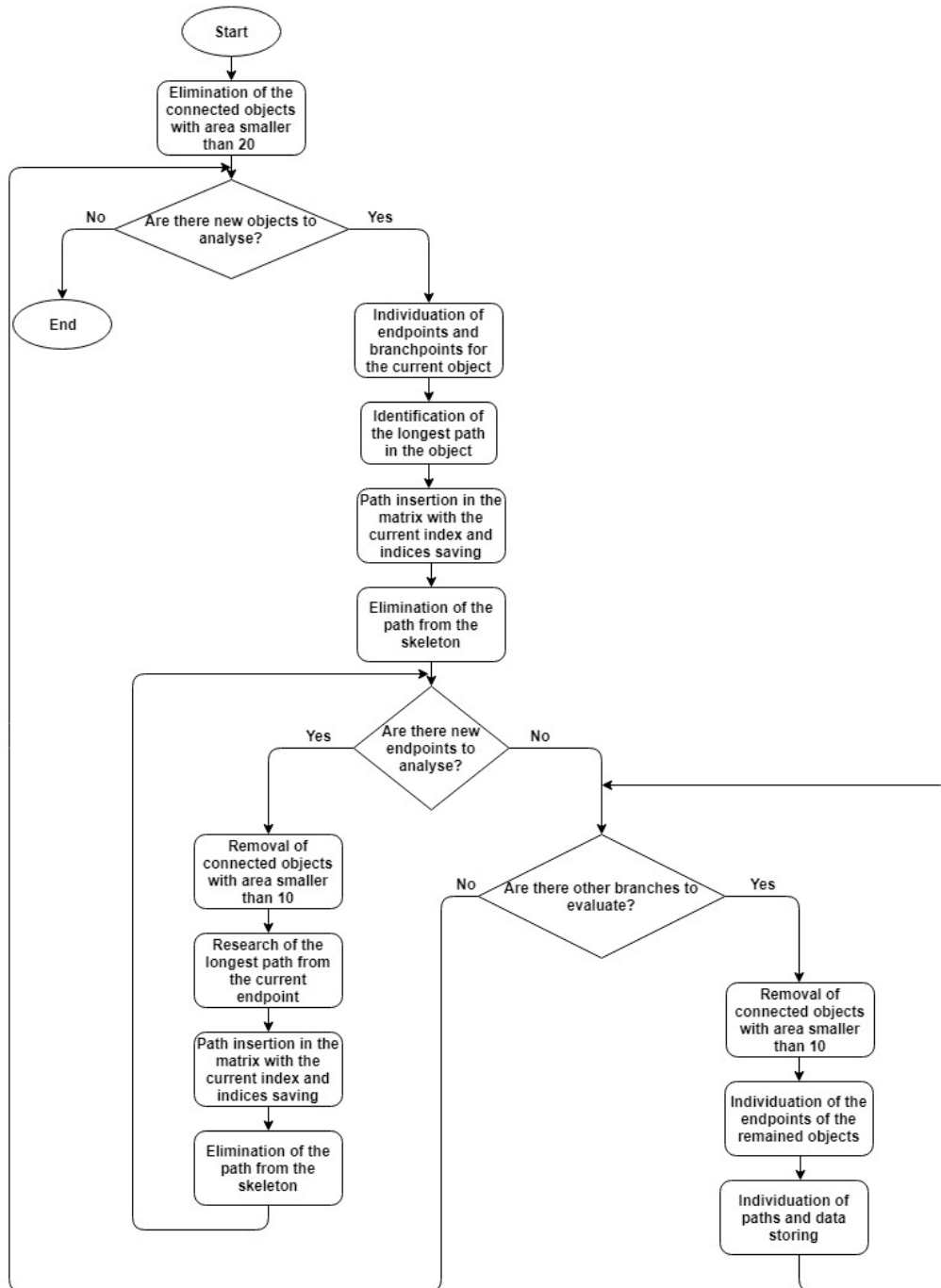


Figure 6.27: Breadth First Search Flowchart.

6.3 Parameters

For the determination of vessels complexity, three tortuosity parameters, three morphological parameters and the mean radius were calculated. The first six were calculated using an already implemented function [23].

6.3.1 Tortuosity Parameters

As already anticipated, the clinical estimation of vascular tortuosity is useful for the diagnosis, monitoring and therapy of a lot of pathologies that affect vessels morphology. For this reason, it's important to develop automatic methods to detect and quantify vessels twisting from 3D medical images. For example, tortuosity is useful to evaluate atherosclerosis seeing as it is connected to risk of stroke and failure of therapies. Moreover some pathologies, such as diabetes and hypertension lead to the formation of small and tortuous vessels, while malignant lesions and malformations cause vessels clusters.[18]

As reported in the Webster's dictionary, the definition of "tortuous" is "marked by repeated twists, bends, or turns, winding, crooked" [55]. But the transformation of the clinician's perception of tortuosity in a defined mathematical metric is absolutely not a trivial problem. Ideally, it would be desirable, an automatic method that detects vessels abnormalities analysing 3D images respect to healthy vasculature. Nevertheless, many researches focused their attention on the tortuosity determination in 2D images, and very few works are available as it regards the 3D evaluation. Moreover, a common limit of the usable information, is the starting assumption that a normal vessel is approximable to a straight line. The most common parameter evaluated in 2D to measure vasculature tortuosity is the "distance metric" that is defined as the ratio between the actual length of the path and the linear distance among the two endpoints of the curve. However, this measure is not sensitive to vessels "wiggles" frequency: because of its definition, this metric associates to a normal and long vessel with a "S" or "U" shape a bigger tortuosity value than to an abnormal vessel with close coils and high frequency oscillations.[18]

Another problem connected to the evaluation of vasculature tortuosity is the definition of "amplitude" and "frequency". Sure enough there are cases such as retinopathy of prematurity where sinusoidal curves are produced, so such definitions are simple, but in other pathologies it isn't so obvious because complex shape vessels appear, such as corkscrew shape or conglomerations. To simplify the problem, some definitions are given: an helix shape is called "coil", the "frequency" is defined as the frequency of a sinusoidal curve or for complex curves as the frequency of a full turn around a cylinder with a unit length axis, lastly the "amplitude" is the amplitude of a sinusoidal curve or the radius of a coil.[18]

There are three common 3D abnormal patterns connected to different pathological processes:

The first type is the simplest and a lot of works about it exist in 2D images. Tortuosity of type I is when a straight or little curved vessel evolves in a "C", "S" or "repeated S" shape because of an elongation. This process is associated to problems such as aging, hypertension, atherosclerosis, retinal diseases or wall vessels hereditary pathologies. Moreover, this abnormality is connected to the possibilities of thrombosis and stroke, and for this reason its quantitative evaluation is clinically interesting.[18]

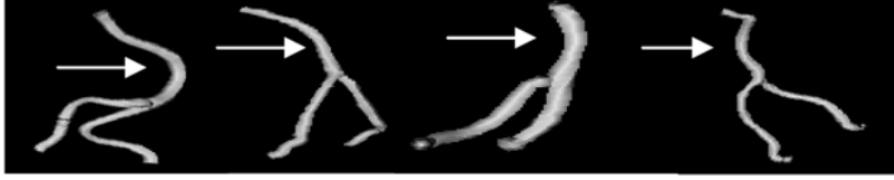


Figure 6.28: Example of type I tortuosity in the basilar artery shown at left. At right, three normal subjects' vessels are reported. This abnormality produces cranial nerve deficits.[18]

The second type is typical of high vasculature tumours and of the arteriovenous malformations nidi (AVMs). The features of this type are the small volume packaging and the frequent and unpredictable oscillations. Moreover, evaluating this parameter it's possible to observe tortuosity reduction due to anti-angiogenic factors so it can be useful for treatment monitoring.[18]

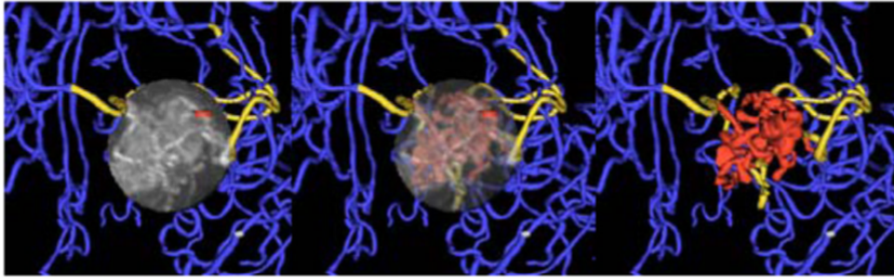


Figure 6.29: Example of type II tortuosity in an AVM nidus. Vessels with different curvatures and lengths are packed in a small volume. Vessels inside the tumour are red, vessels that pass or supply it are yellow and healthy vessels outside it are blue. At left the nidus is shown with a full opacity, in the centre with a partial opacity and at right the vessels are alone.[18]

The third type of abnormality appears in malignant brain tumours. This is characterized by high frequency coils in different length straight or curved vessels. This is useful to identify active grown tumours and to monitor therapies.[18]

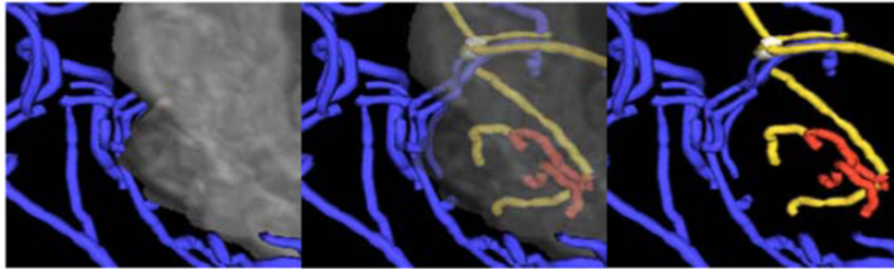


Figure 6.30: Example of type III tortuosity in a malignant glioma. Colours legend is the same of the previous figure. Vessels inside the tumour have high frequency and low amplitude coils or sinusoidal patterns.[18]

Type	Length	Amp	Freq	Comment
I	Long	High	Low	Sinuous curves in long, normally straight vessels
II	Variable	Med	Med	Tightly packed cluster, erratic directional changes
III	Variable	Low	High	Tight coils or sine waves

Table 6.2: Abnormal Tortuosity Types.[18]

The three used tortuosity measures are the “distance metric” (DM), the “sum of angles metric” (SOAM) and the “inflection count metric” (ICM).

For measuring them, vessels medial axis have to be calculated. In fact, the input of the calculations is a set of 3D ordered and regularly sampled points that indicate the spatial position of vessels skeleton.

In the following description, bold font is used for vectors, while italicized font is used for points. Moreover, it’s remembered that a vector is produced by the difference between two points.

Let define “n” as the number of curve points, P_k as the point k of the skeleton, P_0 the first point and P_{n-1} the last one.[18]

Distance Metric (DM)

As it was already mentioned, the distance metric is a dimensionless measure that is defined as the ratio between the actual path length and the linear distance among endpoints. So, the distance metric quantifies how a path deviates from a straight line. The extension to the 3D case is simple.[18]



Figure 6.31: Distance metric defined as the sum of distances between adjacent 3D points along actual vessel path (indicated with short narrow arrows) divided by the straight path between the endpoints (long narrow).[18]

Inflection Count Metric (ICM)

This measure is used in order to overcome the limits of DM, that doesn’t discern among a large “C” vessel and a tortuous one with a lot of oscillations. ICM is a modification of DM where it is multiplied by the number of the “inflection points”.

For a curve in 3D, an inflection point is a minimum of the total curvature. Passing through this point, Frenet frame Normal and Binormal axes change their orientation about by 180° . To identify the 3D inflection points, it’s possible to search the large local maxima of the dot product $\Delta \mathbf{N} \cdot \Delta \mathbf{N}$, where \mathbf{N} is the unit vector of the Frenet normal axis and $\Delta \mathbf{N}$ the difference between normal axes of P_k and P_{k-1} .

The Frenet-Serret formulas define the geometric properties of a 3D continuous and differentiable curve in the Euclidean space \mathbb{R}^3 or the kinematics features of the moving of a particle along this

curve. In particular, they define the derivatives of the tangent, normal and binormal unit vectors, called TNB frame or Frenet-Serret frame that are an orthonormal basis spanning. Precisely, \mathbf{T} is tangent to the curve and point as the motion direction, \mathbf{N} is the ratio between the derivative of \mathbf{T} respect to the arclength of the curve and its length and \mathbf{B} is the cross product between \mathbf{T} and \mathbf{N} . [56]

In Figure 6.32 it's reported a geometric implementation of the Frenet frame. On the left there is a space curve. The velocity vector \mathbf{V} of P_k is approximated to the vector between P_{k-1} and P_{k+1} while the acceleration vector \mathbf{A} is approximated to the subtraction of the vector $\mathbf{T1}$ from $\mathbf{T2}$. On the right, the Frenet frame is derived from velocity and acceleration. \mathbf{T} is the normalized \mathbf{V} , \mathbf{N} is the normalized crossing of \mathbf{V} , \mathbf{A} and \mathbf{V} . \mathbf{T} and \mathbf{N} are in plane and orthogonal. \mathbf{B} is the crossing of \mathbf{T} and \mathbf{N} . [18]

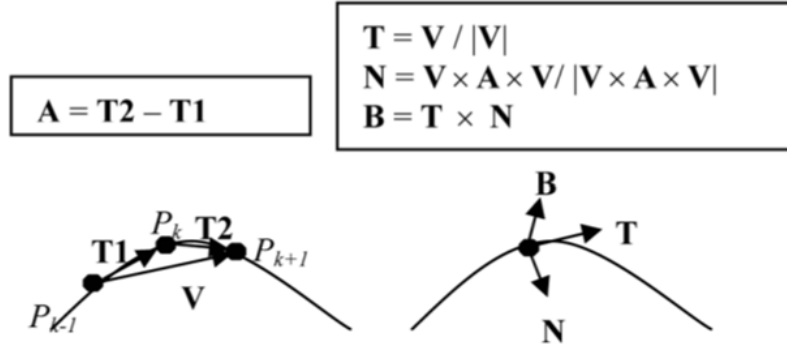


Figure 6.32: Geometric implementation of the Frenet Frame. [18]

If \mathbf{A} has length equal to zero, as for example in inflection points or passing on a straight line, the Frenet frame is undefined. To overcome this problem, if the acceleration length is smaller the 10^{-6} cm, the specific point is skipped and the next is used. [18]

To understand better how to find inflection points, in Figure 6.33 there is an example of a 3D sinusoidal curve with three inflection points (red ovals) and four peaks that are minima or maxima where \mathbf{N} changes its orientation fast, but it doesn't turn upside down. In the image above, it's possible to observe the \mathbf{T} (black narrows) and \mathbf{N} (white narrows) vectors in different points. In particular, in each inflection point, \mathbf{N} rotate of 180° so $\Delta\mathbf{N}$ is about 2. In the figure below, the square length of $\Delta\mathbf{N}$ is reported respect to time. In the inflection points there is a value of 4, in the peaks of 0.01 and in the other locations it is very small. So, if $\Delta\mathbf{N} \cdot \Delta\mathbf{N}$ is greater than 1 there is an inflection point. [18]

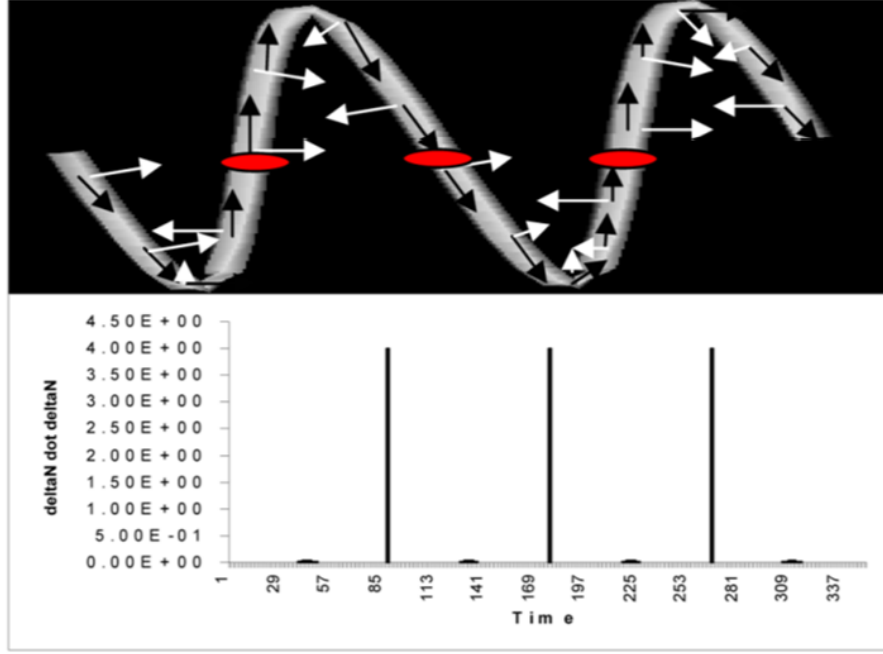


Figure 6.33: Recognition of inflection points in a 3D sine.[18]

Moreover, if there aren't inflection points, ICM is 0 and indicates no tortuosity even if the curve is a large arc. For this reason, 1 is added to the result and in this way, a straight line or a coil have an ICM of 1. So, ICM is always an integer multiple of DM but it is more sensitive to oscillating curves. [18]

Sum of angles metric (SOAM)

Both DM and ICM treat in the wrong manner tight coils. The reason is that coils with high frequency and low amplitude don't add a lot to the total length and that they don't have inflection points, so for DM and ICM the curve is close to a straight line and they report the same small tortuosity value.

So, in these cases, it's necessary to use a parameter that integrates the total curvature along the path and that normalizes it by the length.[18]

As shown in Figure 6.34, for finding the in-plane curvature at a specific point P_k (IP_k), the arccosine of the dot product between the normalized vectors $\mathbf{T1}$ between P_{k-1} and P_k and $\mathbf{T2}$ between P_k and P_{k+1} is calculated in order to obtain the angle among the two vectors. Consequently, if the three points are collinear, the result is 0, if they are nearly collinear the result is small and if they form a sharp peak the result is large:

$$\begin{aligned}
 \mathbf{T1} &= P_k - P_{k-1} \\
 \mathbf{T2} &= P_{k+1} - P_k \\
 \mathbf{T3} &= P_{k+2} - P_{k+1} \\
 IP_k &= \cos^{-1} \left(\left(\frac{\mathbf{T1}}{|\mathbf{T1}|} \right) \cdot \left(\frac{\mathbf{T2}}{|\mathbf{T2}|} \right) \right), IP_k \in [0, \pi]
 \end{aligned}$$

The torsion in a point P_k (TP_k) is defined as the angle among the current osculating circle plane for which the surface normal is the cross product between $\mathbf{T1}$ and $\mathbf{T2}$ and the surface normal of the next osculating plane define by P_k , P_{k+1} and P_{k+2} . The normal vectors of two subsequent osculating planes point in opposite directions if the points P_k and P_{k+1} are in opposite sides of an inflection point. In order to avoid assigning to a planar curve a torsional angle of 180° , in the case of an inflection point, a value of 0 is assigned.[18]

$$TP_k = \cos^{-1} \left(\left(\frac{\mathbf{T1} \times \mathbf{T2}}{|\mathbf{T1} \times \mathbf{T2}|} \right) \cdot \left(\frac{\mathbf{T2} \times \mathbf{T3}}{|\mathbf{T2} \times \mathbf{T3}|} \right) \right), TP_k \in [0, \pi]$$

So, the in-plane and the torsional angles are two positive angles with values among 0 and 180° . For measuring the total curvature at point P_k , CP_k , that is the square root of the sum of the squares of the two angles is made.[18]

$$CP_k = \sqrt{((IP_k \times IP_k) + (TP_k \times TP_k))}$$

Finally the total angles are summed for all the points in the path and then the value is normalized by the total length. So the result is in radians/cm.[18]

$$SOAM = \frac{\sum_{k=1}^{n-3} CP_k}{\sum_{k=1}^{n-1} |P_k - P_{k-1}|}$$

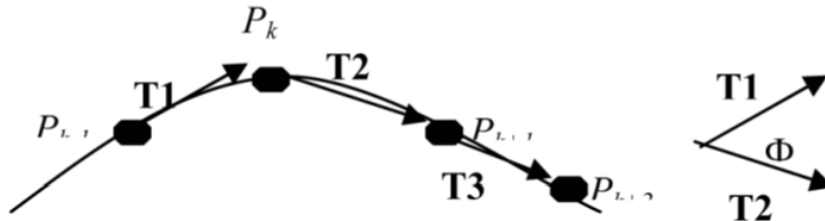


Figure 6.34: In-plane angle Φ for point P_k . [18]

Basing on the defined parameters, given a total path length, high frequency curves must have bigger tortuosity values and for a given frequency a curve with higher amplitude must have a greater tortuosity value.

If the object in analysis is a cluster of vessels, for SOAM it's sufficient to sum angles of each vessel and to divide by the total length of all vessels. The result is the averaged angle per unit length. As it regards DM and ICM, doing the traditional average, small vessels would have the same weight of the long ones. So, it's made the ratio between the sum of all the numerators and the sum of all the denominators, that is the weighted average in order to assign a higher value to longer tortuous vessels. At the end there is one value for each metric for each vascular cluster.[18]

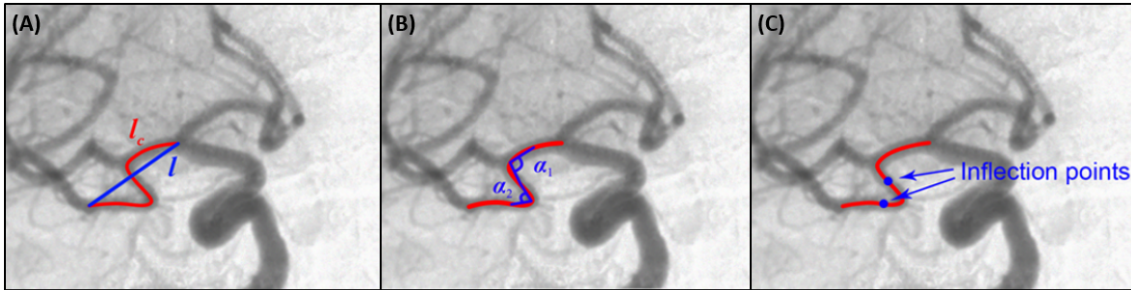


Figure 6.35: Schematic representation of the three tortuosity parameters - (A) DM (B) SOAM (C) ICM.[19]

6.3.2 Morphological Parameters

Apart the tortuosity parameters also three morphological parameters were measured: number of trees (NT) number of branchpoints (NB) and vascular density (VD).

- NT: the number of independent trees identified by the algorithm in the skeleton.
- VD: the ratio between the number of the skeleton voxels and all the voxels of the matrix.
- NB: the number of branchpoints of the structure calculated with the algorithm implemented.[23]

These parameters are useful because it was seen that they are different in the case of healthy or pathological lesions. The number of trees is bigger in the case of tumoral tissues and they increase in time. The same thing is for VD and NB.

Even the radius is different because in a tumoral lesion vessels are small to occupy little space and all the volume.[7]

6.3.3 Mean Radius

To calculate the mean radius of the models, two different methods were tried.

In the first method the function *bwdistgeodesic* is used. At first, model perimeter is found with the function *bwperim*. A point belongs to the perimeter if it is nonzero and it is connected to at least one zero voxel.[57] Then the function *bwdistgeodesic* is exploited to compute the geodesic distance transform between each point of the model and each point of the skeleton. For each true voxel

in the model, this function assigns a number which is the minimum distance between the voxel and the nearest true point in the skeleton. The output value for false voxels is Nan because they can't be traversed in the distance computation, while it is Inf, if there isn't a connected path from seed locations to that element.[58] After this, for finding the radii, only perimeter distances from skeleton with positive and non-infinite values were considered and finally their arithmetic mean and standard deviation were calculated.

The possible options with *bwdistgeodesic*, for computing distance, are “city block”, “chessboard” and “quasi-euclidean”. The last has been chosen because as it is shown in the figure, among all, it best approximates the circular shape of the Euclidean method.[20]

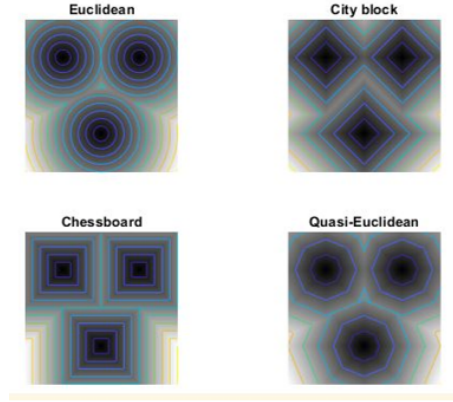


Figure 6.36: Comparison among different distance metrics.[20]

In 2D, the definitions of these distances between the two points (x_1, y_1) and (x_2, y_2) are [58]:

$$\text{'chessboard'} : \max(|(x_1 - x_2)|, |(y_1 - y_2)|)$$

$$\text{'cityblock'} : |(x_1 - x_2)| + |(y_1 - y_2)|$$

$$\text{'quasi-euclidean'} : |x_1 - x_2| + (\sqrt{2} - 1)|y_1 - y_2|, \text{ if } |x_1 - x_2| > |y_1 - y_2| \text{ otherwise : } (\sqrt{2} - 1)|x_1 - x_2| + |y_1 - y_2|$$

The extension to the 3D case is trivial. However, in order to use a more precise measure, a second method was implemented. After finding the perimeter as before, the Euclidean distance was computed between each point of the perimeter and each point of the skeleton.

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Then, for each voxel of the perimeter is found the minimum distance. Last, also in this case, arithmetic mean and standard deviation were calculated.

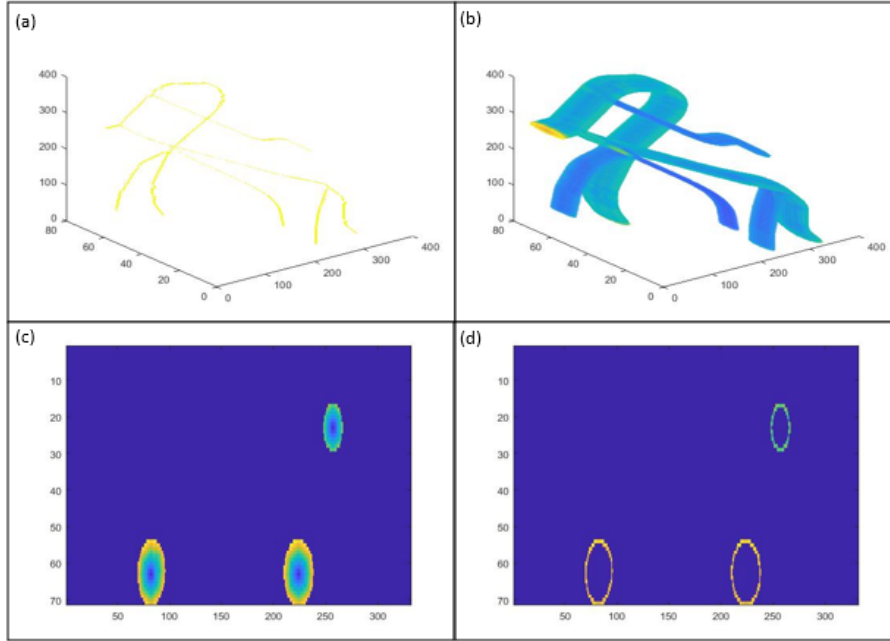


Figure 6.37: (a) Skeleton of M_{simple} , (b) bwdistgeodesic of M_{simple} , (c) Slice 250 of bwdistgeodesic of M_{simple} (d) Slice 250 of perimeter bwdistgeodesic of M_{simple} .

The conversion factor from pixel to mm is different in the three directions (x,y,z). Furthermore, for each point of the perimeter, the nearest skeleton could be in any position. So, making the arithmetic mean and then converting with the averaged conversion factor, an error is committed. To overcome this limit, the weighted mean was also computed considering the radius direction for each point and then the results were compared. This alternative was used only with the second method because it requires to find for each perimeter element, the skeleton point respect to which the radius is computed for knowing the direction of interest, but with the function *bwdistgeodesic* there isn't an option that allows to save the seed considered for the geodesic distance of each specific point. Instead, with the second method, when the minima are calculated, the indexes can be saved.

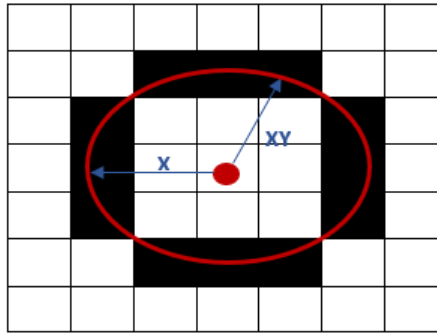


Figure 6.38: Radius value calculated in different points has to be converted differently.

As it regards the acquired images, the conversion factor for x and y directions was found thanks to the scale bar reported in the images. Counting pixels corresponding to 1 mm it was found that for the first and the second dimensions 1 mm corresponds to 16 pixels. The third dimension instead has a conversion factor of 1 because the distance among the acquired slices inserted in the matrix with steps of 1 pixel is 1 mm. The other directions are calculated making the means among these three conversion factors. In the table 6.3 are reported the factors for each direction.

Direction	Conversion Factor
x	16
y	16
z	1
xy	16
xz	8.5
yz	8.5
xyz	11

Table 6.3: Conversion Factor Automatic Models.

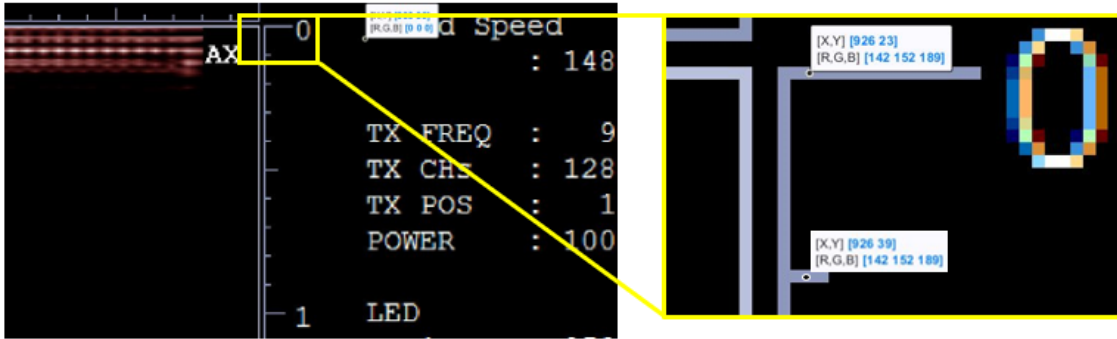


Figure 6.39: Determination of resolution in acquired images.

The conversion factors for the original models depend on the choices done making the transformation from Solidworks to Matlab. As already described, to not deform the models, the same conversion factor has been chosen for all the directions. However, having the matrix sizes to be integer, some approximations are necessary:

For each model the three dimensions in mm are considered. Each value is multiplied by 10 and approximated to the nearest integer. Then to calculate the conversion factor it's sufficient to divide the integer matrix size by the original model size. Even in this case the other dimensions are found making the means.

Direction	Conversion Factor	Direction	Conversion Factor	Direction	Conversion Factor
x	9.97	x	9.97	x	9.99
y	9.99	y	9.99	y	9.99
z	10	z	10	z	10
xy	9.98	xy	9.98	xy	9.99
xz	9.985	xz	9.985	xz	9.995
yz	9.995	yz	9.995	yz	9.995
xyz	9.986	xyz	9.986	xyz	9.993

Table 6.4: Conversion Factor original models - M_{simple} on the left, $M_{complex}$ in the centre, M_{deep} on the right.

As shown in the tables, all the values are very similar to 10, so in the case of the original models it's possible to do the arithmetic mean with a very small error.

The real mean values of the radii were also calculated manually from *Solidworks* in order to validate the algorithm implemented:

M_{simple} : 0,7776 mm

$M_{complex}$: 0,7088 mm

M_{deep} : 0,7086 mm

Chapter 7

Results

In the tables, the results of tortuosity and morphological parameters computation are reported. For each model, there are two different tables: one for each threshold tried for the elimination of smaller skeleton connected objects in the BFS algorithm. As it is shown, with a threshold of 20 better results in terms of DM and ICM are obtained, while with a threshold of 50 the other parameters are more similar to that of the ground truth. However, the reason for which this happens is that with the higher threshold, more pieces of skeleton are removed and so SOAM, NT, NB and VD, that for construction increase with the growth of the skeleton, are obviously smaller, but actually a lot information is lost. For this reason, the next considerations were made using a BFS threshold of 20.

Moreover, inside each table, values were calculated both for automatic and manual segmentation and for each of them two different skeleton cleaning thresholds were used. Manual segmentation results are reported only to have the confirm that automatic segmentation works well, while among the two thresholds of 20 and 40 there are few differences but the first is slightly better and so it was chosen.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	2.1330	2.8816	2.4727	2.1554	2.0883
ICM	86.1439	96.0749	86.3709	62.6852	55.0727
SOAM	0.0373	0.1142	0.2177	0.0910	0.1303
NT	1	6	7	8	9
VD	0.000035410	0.00007795	0.00009393	0.00008573	0.00009811
NB	5	10	19	13	21
MR - Eucl weigh(mm)	0.7526	0.8027	0.7205	0.7125	0.6547
stdR - Eucl weigh(mm)	0.1547	0.6803	0.6763	0.6236	0.6179
MR - Eucl arith(mm)	0.7603	0.7377	0.6443	0.6410	0.5715
stdR - Eucl arith(mm)	0.1561	0.4339	0.3946	0.4133	0.3749
MR - quasi Eucl(mm)	0.81565	0.8106	0.6929	0.6948	0.6133
stdR - quasi Eucl(mm)	0.17101	0.4643	0.3800	0.418	0.3498

Table 7.1: Results M_{simple} - Threshold 20.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	2.1330	3.5585	3.3788	2.3111	2.2489
ICM	86.1439	131.9299	117.2950	72.8454	64.7562
SOAM	0,0373	0,0547	0,1532	0,0776	0,1046
NT	1	2	2	5	5
VD	0.000035410	0.00006932	0.00008357	0.00007846	0.00008911
NB	5	5	13	13	21
MR - Eucl weigh(mm)	0.7526	0.8027	0.7205	0.7125	0.6547
stdR - Eucl weigh(mm)	0,1547	0,6803	0,6763	0,6236	0,6179
MR - Eucl arith(mm)	0.7603	0.7377	0.6443	0.6410	0.5715
stdR - Eucl arith(mm)	0,1561	0,4339	0,3946	0,4133	0,3749
MR - quasi Eucl(mm)	0.81565	0.8106	0.6929	0.6948	0.6133
stdR - quasi Eucl(mm)	0,17101	0,4644	0,38	0,418	0,3498

Table 7.2: Results M_{simple} - Threshold 50.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	2.1640	2.2426	2.1084	2.2887	2.1198
ICM	67.9352	68.6995	63.0030	91.1364	76.5192
SOAM	0.0410	0.1573	0.2706	0.5561	0.5965
NT	1	11	11	13	14
VD	0.000051679	0.00011820	0.00014542	0.00014250	0.00016758
NB	9	34	57	55	77
MR - Eucl weigh(mm)	0.6874	0.9301	0.8231	0.7413	0.6872
stdR - Eucl weigh(mm)	0.1738	0.7866	0.7252	0.6257	0.6213
MR - Eucl arith(mm)	0.6882	0.8439	0.7318	0.6570	0.5909
stdR - Eucl arith(mm)	0.1740	0.4217	0.3524	0.3471	0.3105
MR - quasi Eucl(mm)	0.73773	0.9405	0.8099	0.7378	0.6599
stdR - quasi Eucl(mm)	0.18844	0.4812	0.3998	0.4082	0.3583

Table 7.3: Results $M_{complex}$ - Threshold 20.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	2.1640	2.4251	2.2294	2.5930	2.2889
ICM	67.9352	79.1865	70.1974	113.9714	89.2441
SOAM	0.0410	0.1184	0.2412	0.5080	0.5454
NT	1	6	6	5	6
VD	0.000051679	0.00010971	0.00013693	0.00012827	0.00015391
NB	9	34	57	55	77
MR - Eucl weigh(mm)	0.6874	0.9301	0.8231	0.7413	0.6872
stdR - Eucl weigh(mm)	0.1738	0.7866	0.7252	0.6257	0.6213
MR - Eucl arith(mm)	0.6882	0.8439	0.7318	0.6570	0.5909
stdR - Eucl arith(mm)	0.1740	0.4217	0.3524	0.3471	0.3105
MR - quasi Eucl(mm)	0.73773	0.9405	0.8099	0.7378	0.6599
stdR - quasi Eucl(mm)	0.18844	0.4812	0.3998	0.4082	0.3583

Table 7.4: Results $M_{complex}$ - Threshold 50.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	1.9408	1.3266	1.4726	1.3362	1.5322
ICM	111.7521	13.4494	18.0505	17.8694	22.7301
SOAM	0.0161	0.0379	0.0545	0.0589	0.0640
NT	1	6	6	8	8
VD	0.000050722	0.00005812	0.00006915	0.00008056	0.00009204
NB	4	10	12	20	23
MR - Eucl weigh(mm)	0.7175	0.6450	0.5437	0.5142	0.4594
stdR - Eucl weigh(mm)	0.2575	0.5379	0.4376	0.4581	0.4079
MR - Eucl arith(mm)	0.7054	0.6144	0.5047	0.4690	0.4066
stdR - Eucl arith(mm)	0.2527	0.4589	0.2955	0.3417	0.2397
MR - quasi Eucl(mm)	0.75627	0.6488	0.5298	0.5104	0.4453
stdR - quasi Eucl(mm)	0.27258	0.4605	0.2907	0.3725	0.2757

Table 7.5: Results M_{deep} - Threshold 20.

	Original	Automatic 40	Automatic 20	Manual 40	Manual 20
DM	1.9408	1.3383	1.4904	1.3543	1.5768
ICM	111.7521	13.9502	18.7347	19.7825	25.3070
SOAM	0.0161	0.0340	0.0503	0.0558	0.0612
NT	1	5	5	5	5
VD	0.000050722	0.00005613	0.00006716	0.00007213	0.00008362
NB	4	10	12	20	23
MR - Eucl weigh(mm)	0.7175	0.6450	0.5437	0.5142	0.4594
stdR - Eucl weigh(mm)	0.2575	0.5379	0.4376	0.4581	0.4079
MR - Eucl arith(mm)	0.7054	0.6144	0.5047	0.4690	0.4066
stdR - Eucl arith(mm)	0.2527	0.4589	0.2955	0.3417	0.2397
MR - quasi Eucl(mm)	0.75627	0.6488	0.5298	0.5104	0.4453
stdR - quasi Eucl(mm)	0.27258	0.4605	0.2907	0.3725	0.2757

Table 7.6: Results M_{deep} - Threshold 50.

For analysing tortuosity and morphology results, Figure 7.1 reports for each parameter, values obtained for the three different models respect that of M_{simple} , in the case of the ground truth models (in red) and the automatically reconstructed ones (in black). The aim was to obtain the same trends in the two cases with the purpose to demonstrate that it's possible to characterize vascular complexity from photoacoustic images with the skeletonization approach. Nevertheless, the ink lack in many pieces of the phantoms and the consequent absence of these parts in the acquired images, make this very difficult. In particular, as expected, the most problematic case was that of M_{deep} , for which trends present many differences because only few branches are visible, so a comparison with the ground truth is useless. Instead, as it regards the other two models, trends obtained are very similar to that of the original models, except for the number of independent trees because of disconnections in the acquired images.

However, analysing automatic results respect to the features of the acquired images, it's possible to observe that these trends are perfectly coherent with that models. From data, it emerges that SOAM, NB, NT and VD are bigger for $M_{complex}$ that indeed, is more complex and contains more vessels. Instead DM and ICM are similar because both these parameters are averaged values not additional values, in fact they contain averaged information about the tortuosity of each single vessel and this information is similar for the two models, considering that vessels added to $M_{complex}$ are alike the others. As it regards M_{deep} , the few vessels that are visible don't are so tortuous because they are very long and with few oscillations, so tortuosity and morphological parameters are properly smaller even if ,for the original model, there are some differences because vessels are actually more. So, simply starting from better images where all the vessels are correctly visible, it's possible to obtain correct and useful information about vascular complexity.

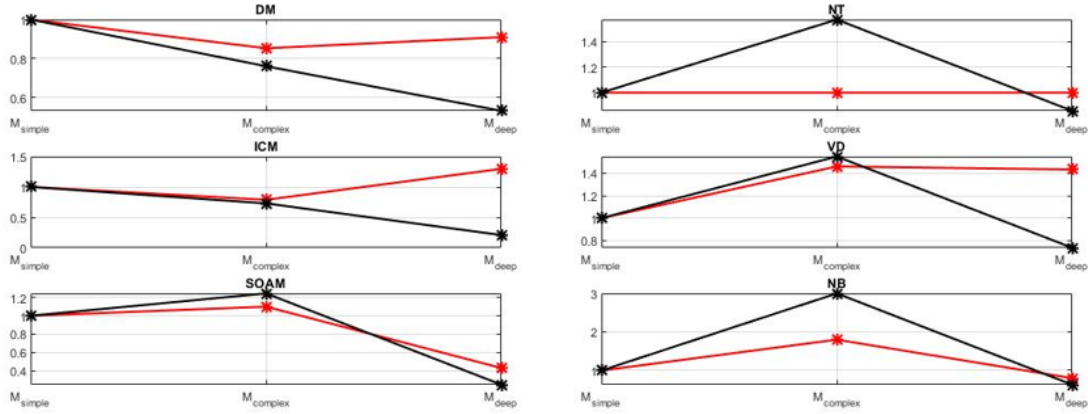


Figure 7.1: Comparison of results among acquired models and ground truth.

As it regards radii, the first histogram reported below (Figure 7.2), shows for each model, the percent error among the different methods tried respect to the real radius value. From this graphic, it's possible to note that among the different distance metrics, the best is the euclidean one, while few differences there are between the weighted and the arithmetical means. So, the second is preferred because it does a little error but has less computational time. In the other histogram (Figure 7.3), results obtained with the chosen method are compared among original models and automatic ones. From these data, it emerges that the mean radius is well estimated, even if for M_{deep} the error is bigger, because in that case few branches are visible.

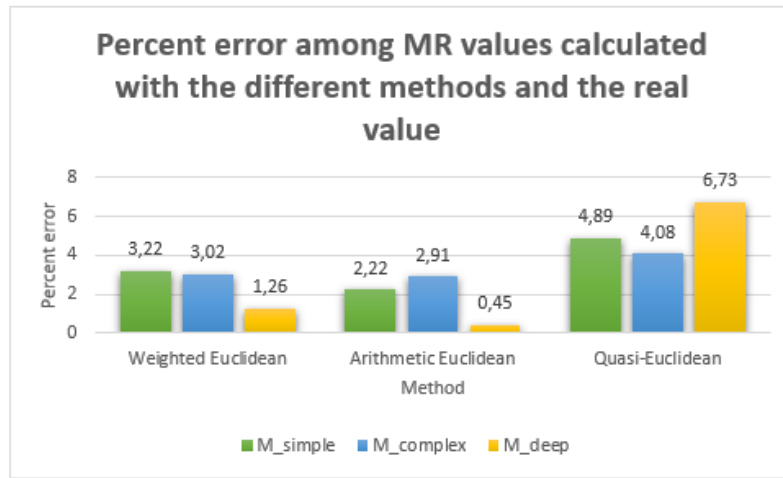


Figure 7.2: Comparison among the different methods for mean radius calculation on the original model respect to the real value calculated manually in Solidworks.

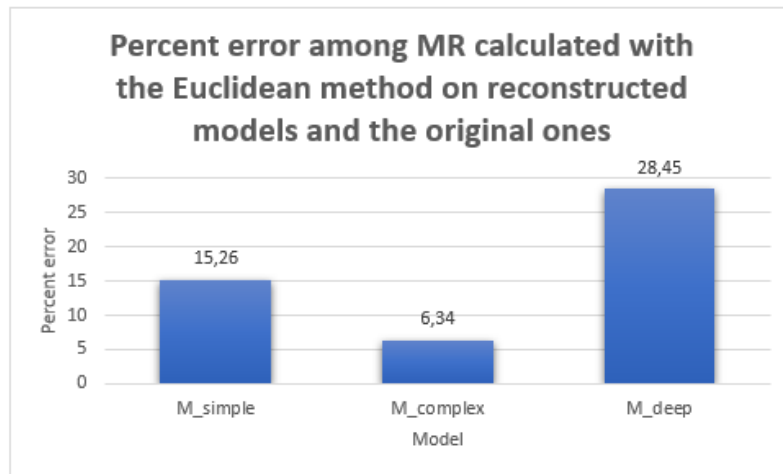


Figure 7.3: Comparison among MR calculated with Euclidean method, on automatic models respect original ones.

Chapter 8

Interpolation

Lastly, for demonstrating the implemented algorithm validity, it was decided to interpolate the skeleton of M_{simple} in order to obtain a unique tree as much as possible similar to the original one. The first model was chosen because, among the three models realised, it is that with the smallest number of missing pieces and so interpolating, a not too much big error is made. For the interpolation the function *interparc*, available online was used. This process was implemented starting from the no-cleaned skeleton obtained from acquired images to have all the usable skeleton pieces. Thereafter, cleaning was made using a threshold of 40, which is a good compromise among the elimination of noise branches and the preservation of shortest real vessels. In the figure 8.1 are reported the MIPs of the original skeleton, the interpolated one and the result after cleaning.



Figure 8.1: M_{simple} - (A) Original skeleton from acquired model (B) Interpolated skeleton MIP (C) Cleaned skeleton MIP.

Tortuosity and morphological parameters were computed on the obtained skeleton with the same procedure described in the previous chapters. Results are reported in the table and the histogram below for comparing them with values obtained from the original model and the automatic reconstruction. It appears immediately clear that in this way there is a big improvement. Obviously, a single tree is found, but as it is shown also tortuosity parameters are closer to real ones. As it regards branchpoints, even for them there is an improvement, but however the number is higher because it's impossible to use a bigger threshold for not deleting real branches of the skeleton. Moreover, it's necessary to remember that interpolating absent vessels, some errors are inevitable because they can't be equal to the real ones.

The aim of this attempt was to confirm the good performance of this algorithm. In fact, starting

from good quality images without missing parts it is perfectly able to provide a very reliable evaluation of vasculature complexity.

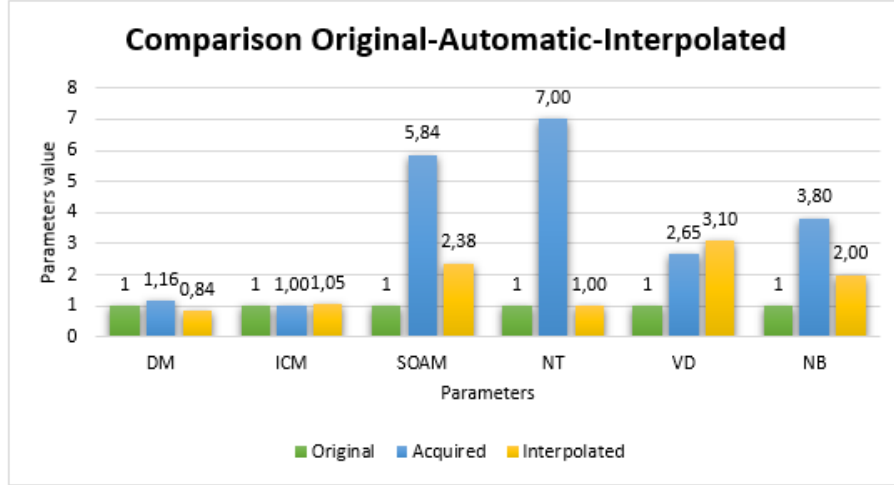


Figure 8.2: Comparison among original skeleton in green, automatic reconstructed skeleton in blue and interpolated skeleton in yellow.

	Original	Automatic	Interpolated
DM	2.1330	2.4727	1.7839
ICM	86.1439	86.3709	90.1441
SOAM	0.0373	0.2177	0.0886
NT	1	7	1
VD	0.000035410	0.00009393	0.00010963
NB	5	19	10

Table 8.1: Comparison among original skeleton, automatic reconstructed skeleton and interpolated skeleton M_{simple} .

Chapter 9

Conclusion and future developments

As already explained, the biggest limitation of the work was the poor quality of acquired images, due to the ink lack in some pieces of the phantoms. This made more difficult the evaluation of algorithm performances because the differences of parameters values respect the ground truth. However, this type of problem can be easily avoided, by realising more precise phantoms. In fact, having enough time to complete the printing procedure, wax can be totally removed and so tubes filling with ink results simpler.

Using the 3D printer is convenient because it is a versatile technique with which it's possible to realise all the desirable shapes and so, several vascular networks with different types of tortuosity can be evaluated. For example, a solution could be starting from simpler vessels with bigger calibres that are easier to fill and then to proceed with a bigger complexity.

Then, also the setup for the acquisition can be improved. For example, it's possible to use a commercial holder that allows to do more stable and precise acquisitions with a higher resolution.

Obviously, it's even possible to improve the algorithm, enhancing as soon as possible the segmentation, considering all possible photoacoustic artifacts, but especially the steps for obtaining the final skeleton. In particular, a huge breakthrough would be the development of an automatic method for the solving of under and over skeletonization and the interpolation of interrupted vessels.

The next step would be the application of this technique to in vivo images with the necessary algorithm improvements for an efficient evaluation.

Despite these limitations, from the obtained results, it was observed that starting from photoacoustic images, the evaluation of vascular complexity using a skeletonization algorithm is possible. In fact, analysing data, it emerges that trends for M_{simple} and $M_{complex}$, where enough branches where visible are similar to ground truth and moreover, even if there are some differences, parameters calculated on reconstructed models follow exactly the features of models to which are referred, obviously only the visible parts. So, starting from good images, this approach is very useful to characterize blood vessels.

Clearly, for applying these measurements to clinical practice, a big statistical analysis is necessary in order to obtain typical parameters values for healthy and different pathological tissues, since there isn't still enough information about it.

In conclusion, it's therefore possible to assert that, the automatic and quantitative characterization of tortuosity and morphological parameters starting from photoacoustic images is a promising and powerful instrument for many clinical applications.

Bibliography

- [1] P. Burgholzer, H. Grun, A. Sonleitner, “Photoacoustic tomography: Sounding out fluorescent proteins” in *Nature Photonics*, v. 3, pp. 378–379, July 2009.
- [2] P. Beard, “Biomedical photoacoustic imaging” in *INTERFACE FOCUS*, v. 1, August 2011.
- [3] [Online]: <http://www.yfp.elisacorteggiani.com/spettrofotometro.html>
- [4] A. Hariri, J. Lemaster, J. Wang, A. Jeevarathinam, D. Chao, J. Jokerst, “The characterization of an economic and portable LED-based photoacoustic imaging system to facilitate molecular imaging” in *Photoacoustics*, v. 9, November 2017.
- [5] T. Agano, N. Sato, H. Nakatsuka, K. Kitagawa, T. Hanaoka, K. Morisono, Y. Shigeta, C. Tanaka, “High frame rate photoacoustic imaging using multiple wave-length LED array light source” in *SPIE*, March 2016.
- [6] [Online]: <http://www.cee.hw.ac.uk/hipr/html/thin.html>
- [7] F. Molinari, *Elaborazione di immagini mediche*, 2017.
- [8] [Online]: <https://www.researchgate.net/figure/a-Left-fundus-image-DRIVE-set-a-right-binary-vasculature-map-b-Left-fig4-44642754>
- [9] C. Kirbas, F. Quek, “Vessel extraction techniques and algorithms : A survey” in *Conference paper*, pp. 238–245, April 2003.
- [10] L. Tan, J. Jiang, *Digital Signal Processing (Second Edition)* Academic Press, 2013.
- [11] S. Moccia, E. De Momi, S. El Hadji, L. Mattos, “Blood vessel segmentation algorithms- Review of methods, datasets and evaluation metrics” in *Computer methods and programs in biomedicine*, May 2018.
- [12] E. Ficarra, *Bioinformatics*, 2018.
- [13] [Online]: <https://it.3dsystems.com/3d-printers/plastic#multijet-printers-mjp>
- [14] [Online]: <http://emlab.utep.edu/ee5390em21/Lecture%2019%20-%20Interfacing%20MATLAB%20with%20CAD.pdf>
- [15] K. Subburaj, S. Patil, B. Ravi, “Voxel-Based Thickness Analysis of Intricate Objects” in *International Journal of CAD/CAM*, v. 6, pp. 105–115, January 2006.
- [16] S. Patil, B. Ravi, “Voxel-based representation, display and thickness analysis of intricate shapes” in *Proceedings - Ninth International Conference on Computer Aided Design and Computer Graphics, CAD/CG 2005*, p. 6 pp., January 2006.
- [17] T. Lee, R. Kashyap, C. Chu, “Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms” in *CVGIP: Graphical Models and Image Processing*, v. 56, pp. 462–478, November 1994.
- [18] E. Bullitt, G. Gerig, S. M. Pizer, W. Lin, S. Aylward, “Measuring Tortuosity of the Intracerebral Vasculature From MRA Images” in *IEEE transactions on medical imaging*, v. 22, pp. 1163–71, October 2003.
- [19] K. Klis, R. Krzyzewski, B. Kwinta, K. Stachura, M. Moskale, K. Tomaszewski, “Computer-aided analysis of middle cerebral artery tortuosity: Association with aneurysm development”

- in *Journal of Neurosurgery*, v. 130, pp. 1–7, May 2018.
- [20] [Online]: https://it.mathworks.com/help/images/ref/bwdist.html?searchHighlight=bwdist&s_tid=doc_srchttitle
 - [21] E. Bullitt, K. Muller, I. Jung, W. Lin, S. Aylward, “Analyzing attributes of vessel populations” in *Medical Image Analysis*, v. 9, pp. 39–49, March 2005.
 - [22] E. Bullitt, S. Aylward, T. Van Dyke, W. Lin, “Computer-Assisted Measurement of Vessel Shape from 3T Magnetic Resonance Angiography of Mouse Brain” in *Methods (San Diego, Calif.)*, v. 43, pp. 29–34, October 2007.
 - [23] K. Meiburger, S. Nam, E. Chung, L. Suggs, S. Emelianov, F. Molinari, “Skeletonization algorithm-based blood vessel quantification using in vivo 3D photoacoustic imaging” in *Physics in Medicine and Biology*, v. 61, pp. 7994–8009, October 2016.
 - [24] S. Shelton, Y. Lee, M. Lee, E. Cherin, F. Foster, S. Aylward, P. Dayton, “Quantification of Microvascular Tortuosity during Tumor Evolution Using Acoustic Angiography” in *Ultrasound in Medicine and Biology*, v. 41, April 2015.
 - [25] J. Perry, J. Statler, “Advances in Vascular Imaging” in *Surgical Clinics of North America*, v. 87, October 2007.
 - [26] Y. Zhang, H. Hong, W. Cai, “Photoacoustic Imaging” in *Cold Spring Harbor protocols*, August 2011.
 - [27] M. Xu, L. Wang, “Photoacoustic imaging in biomedicine” in *Review of Scientific Instruments - REV SCI INSTR*, v. 77, April 2006.
 - [28] K. Meiburger, *Quantitative Ultrasound and Photoacoustic Imaging for the Assessment of Vascular Parameters* Springer, 2016.
 - [29] (1993) Council directive 93/42/EEC on medical devices. [Online]: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1993L0042:20071011:EN:PDF>
 - [30] B. Preim, S. Oeltze-Jafra, “3D Visualization of Vasculature: An Overview” in *Visualization in Medicine and Life Sciences*, pp. 39–59, January 2008.
 - [31] A. Sarwal, A. Dhawan, “3-d reconstruction of coronary arteries” in *IEEE Conf. Eng.in Medicine and Bio.*, v. 1, pp. 504–505, 1994.
 - [32] M. Chwialkowski, Y. Ibrahim, F. Hong, R. Peshock, “A method for fully automated quantitative analysis of arterial flow using flow-sensitized mr images” in *Comp.Med. Imaging and Graphics*, v. 20, pp. 365–378, 1996.
 - [33] T. Tozaki, Y. Kawata, N. Niki, H. Ohmatsu, N. Moriyama, “3-d visualization of blood vessels and tumor using thin slice ct” in *IEEE Nuclear Science Symposium and Medical Imaging Conference*, v. 3, pp. 1470–1474, 1995.
 - [34] E. Bullitt, S. Aylward, “Analysis of time-varying images using 3d vascular models” in *Proc. Applied Imagery Pat. Recog. Works.*, pp. 9–14, October 2001.
 - [35] G. Arce, J. Bacca, J. Paredes, *The Essential Guide to Image Processing* Academic Press, 2009.
 - [36] [Online]: https://en.wikipedia.org/wiki/Otsu%27s_method
 - [37] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms” in *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, pp. 62–66, 1979.
 - [38] [Online]: http://homes.di.unimi.it/ferrari/ImgProc2011_12/EI2011_12_16_segmentation_double.pdf
 - [39] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, R. Kikinis, “3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images” in *IEEE Medical Image Analysis*, v. 2, pp. 143–168, June 1998.
 - [40] A. Hoover, V. Kouznetsova, M. Goldbaum, “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response” in *IEEE Trans. on Med. Img.*, v. 19, pp. 203–210, March 2000.

- [41] S. Eiho, Y. Qian, "Detection of coronary artery tree using morphological operator" in *IEEE Computers in Cardiology*, v. 24, pp. 525–528, 1997.
- [42] R. Chan, W. Karl, R. Lees, "A new model-based technique for enhanced smallvessel measurements in x-ray cine-angiograms" in *IEEE Trans. on Med. Img.*, v. 19, pp. 243–255, March 2000.
- [43] P. Summers, A. Bhalerao, "Derivation of pressure gradients from magnetic resonance angiography using multi-resolution segmentation" in *Proceedings of International Conference on Image Processing and its Applications*, pp. 404–408, 1995.
- [44] C. Smets, G. Verbeeck, P. Suetens, A. Oosterlinck, "A knowledge-based system for the delineation of blood vessels on subtraction angiograms" in *Pattern Rec. Let.*, v. 8, pp. 113–121, September 1988.
- [45] W. Vogt, C. Jia, K. Wear, B. Garra, T. Pfefer, "Biologically relevant photoacoustic imaging phantoms with tunable optical and acoustic properties" in *Journal of biomedical optics*, v. 21, p. 101405, February 2016.
- [46] M. Fonseca, B. Zeqiri, P. Beard, B. Cox, in "Characterisation of a PVCP based tissue-mimicking phantom for Quantitative Photoacoustic Imaging" p. 953911, January 2015.
- [47] S. Bohndiek, S. Bodapati, D. Van de Sompel, R. Kothapalli, S. Gambhir, "Development and Application of Stable Phantoms for the Evaluation of Photoacoustic Imaging Instruments" in *PloS one*, v. 8, p. e75533, September 2013.
- [48] J. Cook, R. Bouchard, S. Emelianov, "Tissue-mimicking phantoms for photoacoustic and ultrasonic imaging" in *Biomedical optics express*, v. 2, pp. 3193–206, November 2011.
- [49] L. Cabrelli, D. Sampaio, J. Uliana, A. de Ana, A. A. Carneiro, T. Pavan, in "Copolymer-in-oil phantoms for photoacoustic imaging" October 2015.
- [50] Y. Guo, H. Patanwala, B. Bognet, "Inkjet and inkjet-based 3D printing:connecting fluid properties and printing performance" in *Rapid Prototyping Journal*, pp. 562–576, June 2017.
- [51] [Online]: https://www.amazon.it/Polvere-Qualit%C3%A0-Etichette-istruzioni-Italiane/dp/B0792FTYCQ/ref=sr_1_3?keywords=agar+agar+polvere&qid=1556997136&s=gateway&sr=8-3
- [52] [Online]: <https://it.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation>
- [53] H. Homann, in "Implementation of a 3D thinning algorithm" January 2007.
- [54] J. Silvela, J. Portillo, "Breadth-first search and its application to image processing problems" in *Image Processing, IEEE Transactions on*, v. 10, pp. 1194–1199, September 2001.
- [55] [Online]: <https://www.merriam-webster.com/dictionary/tortuous>
- [56] [Online]: https://en.wikipedia.org/wiki/Frenet%E2%80%93Serret_formulas
- [57] [Online]: https://it.mathworks.com/help/images/ref/bwperim.html?searchHighlight=bwperim&s_tid=doc_srchttitle
- [58] [Online]: https://it.mathworks.com/help/images/ref/bwdistgeodesic.html?s_tid=doc_ta

Acknowledgments

In conclusion of this thesis, I consider very important to thank all those who support me.

Firstly, my supervisors, the professors Filippo Molinari and Kristen Meiburger for giving me the possibility to undertake this thesis work and to face it, giving me constant support and assistance with high availability and professionalism.

Secondly, Alessandro Sanginario, the professor Alberto Vallan and his student Vittorio for their help during my work.

Then, I want to thank my mother Giusy, my father Massimo, my sister Giorgia and my grandmothers Mafalda and MariaTeresa for the big moral and economic support. Without them, all my university career would not have been possible.

A big gratitude is also for my boyfriend Simone, for the patience and the love with which he has been near me in this difficult path.

I thank so much my flat mates and friends Anna and Alessandra that shared with me every moment of my university experience, each happiness and each suffering.

Moreover, I have to thank my friend Luca for his kind and essential help.

Obviously, I can't forget all the people that I knew in these years, the colleagues with which I have worked together and the friends who, with their presence, made this period less difficult.

Last but not least, I thank my sweet dog Kira, that kept me company during all my thesis writing.