Real-time Haptic Guidance System for Retinal Surgery Based on Intraoperative Optical Coherence Tomography

BY

MATTEO DE SILVESTRI
B.S, Politecnico di Milano, Milan, Italy, 2016

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Bioengineering
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:

Cristian J. Luciano, Chair and Advisor

Xincheng Yao

Yannel Leiderman, Ophthalmology

Marco D. Santambrogio, Politecnico di Milano

ACKNOWLEDGMENTS

ACKNOWLEDGMENTS (continued)

Thank you to Riccardo, the best roommate I could have wished to have: you made good times better, and hard times easier. Thank you to Eleonora, for being the perfect coffee mate during the months spent together.

To my friends of a lifetime, waiting for me in Italy: thank you guys for your incredible ability to make me feel at home, even when an ocean was keeping us apart.

Finally, the greatest thank you goes to my family: my parents Marco and Laura, principal sponsor of this whole adventure, my grandmother Giulia, my sister Barbara and my (eventually) brother in law Mattia. Thank you for always supporting my choices and for giving me all the advice and love you have.

MDS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

OCT	Optical Coherence Tomography
VL	Visualization Library
QH	QuickHaptics™
AS	Ascension
WK	Wykoby Computational Geometry Library©
VR	Virtual Reality
AR	Augmented Reality
UIC	University of Illinois at Chicago
API	Application Programming Interface
BSD	Berkley Source Distribution
TD-OCT	Time Domain Optical Coherence Tomography
SD-OCT	Spectral Domain Optical Coherence Tomography
FD-OCT	Fourier Domain Optical Coherence Tomography
OFDR	Optical Frequency Domain Reflectometry
SNR	Signal to Noise Ratio

LIST OF ABBREVIATIONS (continued)

OpenCV	Open Source Computer Vision Library
OR	Operating Room
iOCT	Intraoperative Optical Coherence Tomography
RD	Retinal Detachment
ERM	Epiretinal Membrane
PVR	Proliferative vitreoretinopathy
PVD	Posterior Vitreous Detachment
IRISS	Intraocular Robotic Interventional Surgical System
DOF	degrees of freedom
MRF	Markov Random Fields
RFC	Random Forest Classifier
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
AABB	Axis Aligned Bounding Box

SUMMARY

Ophthalmic surgical procedures are hand-performed procedures in which the surgeons operate tools at close contact with the patient eyes delicate tissues. The range of movement is generally in the order of hundreds of micrometers, and no tactile information is provided, given the low mechanical properties of the involved tissues (retinal layers, vitreous humor, epiretinal membranes, etc.). In most cases the outcomes of these procedures are strictly correlated to factors – such as the amount of force applied to the tissues or the proximity to certain critical structures – which are difficult to objectively evaluate. Visual feedback is the only relevant information supporting the surgeon's experience, which therefore plays a fundamental role in keeping all those variables under control. Computer-aided analysis of the images generated by already-in-use intraoperative monitoring methods has the potential to provide additional objective information to the physician. This information can be integrated in a computing system to guide the surgeon, or to output real-time evaluation of the procedure.

This work proposes an image-based guidance provided by means of haptic feedback to the surgeon. As it produces the desired force effect, the haptic device can track the operating tools. This information can be integrated with the information extracted in real-time from the scans of the retina to create a controlled environment for surgery, with a significant perception enhancement.

CHAPTER 1

BACKGROUND

Retinal surgery procedures, such as epi-retinal membrane peeling [1] or retinal detachment repair [2], require the surgeon to operate in extremely low scale environment. Unwanted movements of less than a millimeter can prejudice the outcomes of the operation, if not even injure the patient. Years of training provide the surgeon with the right amount of experience to deal at best with each situation, supported by many intraoperative monitoring instruments. However, objective evaluation measures of the ongoing procedure are still missing, with this potentially leading to suboptimal results.

OCT [3] is a non-invasive imaging technique based on the interferometer working principle. Specific light wavelengths are deployed according to the desired resolution, which is generally related to the field of application. It found a great relevance in ophthalmology, due to the possibility to capture cross-sections of the retina showing all its layers well differentiated. Recently, the development of real-time OCT scanners [4] integrated with operating microscopes has allowed continuous imaging of the retina during the operation. Such a tool has the potential to provide the surgeon with a substantial amount of information. Currently though only the raw visual information is presented to the user, lacking any additional processing.

The project proposes a novel method for providing guidance to surgeons during retinal surgery procedures. Intraoperative monitoring images are analyzed to extract information about the relative position between the operating tools and the retina's delicate tissues. The

following sections aim to give some contextual knowledge, hoping to facilitate a better understanding of the encountered problems and the proposed solutions.

1.1   Vitreoretinal surgery

The branch of ophthalmology called **Vitreoretinal surgery** deals with retinal diseases that needs to be addressed surgically, and consists of both invasive and non-invasive procedures. The introduction of operating microscopes [5] was fundamental for the early applications, and ophthalmology was in fact the first branch of medicine in which they were used.

Below, a brief description of the eye anatomy is followed by an introduction to some of the most common diseases addressed by vitreoretinal surgery.

1.1.1   Anatomy of the eye

The eye is the human organ responsible for light sensing, and the first element of the complex pathway leading to vision. Despite being small – about $2.5cm$ (1") in diameter – its anatomy is complex. A schematic representation is given in Figure 1. The outer portion, referred to as **anterior segment**, is exposed and in charge of adjusting the amount and the direction of the entering light. A highly vascularized tissue called *sclera* wraps the whole organ, except for the central region of the anterior segment. Here the *cornea* allows light to pass through with very few absorption. The *iris*, the colored region of the eye, is a diaphragmatic membrane which controls a variable aperture, the *pupil*, by means of contractions (miosis) and relaxations (mydriasis). This adjusts the amount of light reaching the inside of the eye. Right behind the pupil lies the *lens*, which focuses the light toward the *fovea*. The convergence index of the lens varies together with its shape. This changes due to contractions of the *ciliary bodies*, small

muscles which anchor the lens to the eye walls. The *anterior chamber* and *posterior chamber* are the gaps between the cornea and the iris and between the iris and lens respectively. The layer beneath the sclera is called *choroid*, and it is responsible for the oxygen support of the whole eye. The posterior segment is composed by everything that is behind the lens. The *vitreous body* is the cavity which occupies the most of the ocular volume, and it is filled with a gel-like fluid called *vitreous humor*. The latter is the main factor determining the intra-ocular pressure, and is also an important actor in some of the most common pathologies.

The innermost layer of the eye is the **retina**. It is an extrusion of the nervous system, and it is its most exposed part. The retinal structure is incredibly complex and can be divided in many substructures, each one with its specific purpose. The main one is, of course, the receptors' layer. Two types of receptors are in charge of sensing the incoming light: **rods** and **cones**. The first, more sensitive, are responsible for night vision, and cannot distinguish colors. The second, less sensitive, are deputed to light vision and color perception. Cones are much denser in the central region of the retina (the fovea) while the rods are more concentrated in the areas responsible for peripheral vision. Those receptors form the first layer beneath the choroid. The intricate and complex structure of neural interconnections that covers them contains cells with many different tasks which perform the very first processing of the sensed image. Among the others, *vertical cells* carry the neural signal generated by the receptors toward the central nervous system, whereas the *horizontal cells* transport information laterally, integrating signals arriving from different receptors. Another significant layer of the retina is constituted by the

Figure 1: Sagittal cross section representation of the anatomy of the eye. Image downloaded from: *http://www.laramyk.com/resources/education/occular-anatomy/major-ocular-structures* in April 2018

vascular system, which has the fundamental role of carrying oxygen and supplies to all those cells [6].

1.1.2   Retinal detachment

Retinal Detachment (RD) is the condition in which a portion of the retina separates from the underlining tissue, called choroid. It is particularly widespread in the myopic population, as it is generally caused by traction of the retina by the **vitreous humor**. The vitreous shrinks physiologically with aging as it looses collagen, in a process called **Posterior Vitreous Detachment (PVD)**. In the myopic eye, whose axial length is bigger than normal, this can more easily lead to retinal detachment, but it can happen to healthy eyes as well .

Generally the causes leading to RD are classified as either traumatic or non-traumatic. Traumatic causes [7] are those in which the detachment arises from eye contusions, generally following retinal breaks. PVD is related to aging, whenever RD is due to it, that's a non-traumatic cause.

When it gets detached, the retina can tear apart, allowing the flow of liquids beneath its surface. This causes a bigger portion of the neural tissue to detach, in a positive feedback self-inductive loop. The first symptoms can be related to the release of some blood inside the vitreous, resulting in the patient seeing floating objects within the eye. If neglected, it can cause partial blindness [8].

There are multiple, currently available ways to treat this pathology. The main ones are listed and briefly described below:

1. **Cryotherapy:** a small, freezing probe is put in contact with the external surface of the eye, the *sclera*, in correspondence to where the retina is damaged. The frozen tissue will form a scar, that acts as an adhesive point for the detached retina. The main advantage with this procedure is that it is completely non-invasive; however, an obvious limit is related to the need of reaching the interesting zone from outside, which is not always possible.

2. **Scleral Buckle:** a silicone band is wrapped around the sclera, pushing the inner layers of the eye inward to get them in contact with the retina again. It is an elegant, low invasive solution, whose main side-effects are the discomfort patients claim to feel for a few weeks after the operation, and the possible change in visual acuity.

3. **Pneumatic Retinopexy:** a gas bubble is blown inside the eye posterior chamber. It pushes the torn or detached retina against the eye walls, so that it can be sealed taking advantage of one of the other solutions (e.g., cryotherapy).

4. **Vitrectomy:** a completely different approach. A thin needle, called *vitrectome*, is inserted inside the eye and it is used to remove the vitreous which is pulling the retina. The removed fluid is replaced by some liquid (or gas) which will be slowly reabsorbed by the eye and substituted with physiological fluid. It belongs to the class of minimally invasive, micro-surgery procedures.

1.1.3 Epiretinal membrane

Also known as Macular Pucker, **Epiretinal Membrane (ERM)** is the growth of a tissue layer on top of the inner retina in proximity of a region called **macula**. It might interfere with central vision, causing *metamorphopsia* – vision distortion – or even painless loss of vision. These fibrocellular membranes are avascular and semi-translucent, and they origin from the migration and abnormal growth of glial cells. This might happen for no particular reason – *idiopathic* ERM – or as consequence of PVD, such as for the previously described RD (see 1.1.2) [9].

In many cases this condition is asymptomatic and does not require any intervention. In other circumstances, the preferred way to treat it is via **membrane peeling**. During this procedure the fibrocellular membrane is cut and peeled using ad hoc surgical tools. Everything is performed inside the eye, with the aid of imaging tools and endoscopic lights. Vitrectomy

(see 1.1.2) is used to effectively remove the membrane and ensure the proper healing of the retinal tissue [10].

1.1.4   Proliferative vitreoretinopathy

Proliferative vitreoretinopathy (PVR) is the most common complication that might follow the surgery intervention for RD. In 75% of the cases, it might require repeated surgery for the patient. The symptoms are similar to those of ERM (see 1.1.3), as it consists in the formation of a semi-translucent membrane on the retina inner surface. The fundamental difference is that, in this case, the membrane is some scar tissue, grown after the unwanted damage of the retina. Given the similarities, the intervention procedure is the same as for treating ERM: vitrectomy is performed in order to ease the subsequent peeling of the membrane [11].

According to literature [12], the initial classification was of four stages: severity going from *minimal PVR* to *massive PVR* along stages A, B, C and D. More recently [13], a new classification distinguishes three main grades A, B and C. The third grade is subdivided again into two types: Ca when the membrane grows anteriorly to the equator and Cb when it grows posteriorly to the equator.

1.2   Optical Coherence Tomography

Optical Coherence Tomography is a non-invasive imaging technique which deploys coherent light to get high resolution cross-sections of light-scattering materials. It was firstly introduced in 1991 by Huang et al. [3] as a novel imaging method for both transparent and non-transparent media [14]. An intuitive yet imprecise analogy to understand how it works is to think of it as of a light-based ecography B-mode scan. The working-principle is similar, since

the device correlates the delay of backscattered light to the spatial distance of the reflecting surface. An example of what this imaging technique can create is provided in Figure 3. Besides the different physical entity exploited, the main difference – and one of the main reasons that determined its success – is the finer **axial resolution**.

Spatial resolution is defined as the minimum distance between two contiguous points necessary for them to be distinguished as separated; the smaller the distance, the better the resolution. Axial resolution specifically refers to the spatial resolution in the transversal direction with respect to the scanned surface. From their first implementations, OCT scanners were able to provide resolutions between $10\mu m$ and $1\mu m$, which means up to two order of magnitude smaller then the common ultrasounds scanners. This, though, is paid with a significantly lower penetration depth. Moreover, such small spatial distances reflect into small time delays, which means that a completely different setup is needed to measure them. More in detail, the system used to compute such fine measurements is called **Michelson interferometer** [15].

A schematic representation of an interferometer is shown in Figure 2. A half-silvered mirror works as a beam-splitter (BS), dividing the incident light coming from the source into two rays with equal phase. The reference mirror moves along the direction of traveling of light. The target (sample) is a light scattering material. The two reflected rays rejoin at the beam-splitter, and their sum is sensed by the detector. The interference pattern read by the detector depends on the phase difference between the two backscattered light rays. Since the initial phase is the same, and the light frequency does not change, the phase shift is related only to a different time-of-flight, i.e., a different path length. Similar phase means constructive interference, different

Figure 2: Michelson optical fiber interferometer scheme, with descriptive examples of long and short coherence light. *Image courtesy of Fujimoto et al.*

phase means destructive interference. If light was a stationary sinusoid, there will be periodic constructive interference for multiple distance differences. Low coherence light (see Figure 2) is preferred, because only phase shifts within the range of the coherence length are able to produce constructive interference and be detected. An exhaustive mathematical description can be found in [16].

As the reference mirror moves away from the beam-splitter, let's say at distance d, only light reflected from the target at the same distance d can produce constructive interference. A single tomographic line (A-scan) can be obtained by moving the reference mirror along a line. If multiple adjacent lines are scanned, a 2D image con be reconstructed. The final result will

be a grey-scale image, with the pixels' intensity mapped with the light scattering properties of each point of the target sample.

The system briefly described above refers to the very first implementation of the OCT scanner, and is also known as Time Domain Optical Coherence Tomography (TD-OCT). The depth scanning is performed by mechanically moving the reference mirror, thus resulting in long acquisition times: common TD-OCT can record hundreds of A-scan/sec, which means that they need to be realigned when reconstructing the B-scan, in order to have consistency between the pixels [17].

1.2.1 Spectral-domain OCT

A few years after the first introduction of TD-OCT in 1991, a new OCT reconstruction method was proposed, extracting depth information from the spectral content of the interference pattern. **Spectral Domain Optical Coherence Tomography (SD-OCT)** revolutionized OCT imaging, allowing up to $50\times$ acquisition time speed-up factors with respect to the common time domain methodology [18].

In this new setup the reference mirror is fixed: the depth information which was extracted out of the position of a mechanical component, is now mathematically computed from an instantaneous spectral distribution. Two main techniques can be distinguished:

- **Fourier Domain Optical Coherence Tomography (FD-OCT):** also known as *spectral radar*, it deploys broadband light signals, and the detector is replaced with a low-loss spectrometer. Wide band and low coherence light ensure high axial resolution of the resulting B-scan image [19][18].

Figure 3: Sample image obtained from a commercially available OCT scanner. This kind of imaging is used in diagnostics and screening, exploiting its high definition to visualize effectively all the retinal layers. Image downloaded from: *https://commons.wikimedia.org/wiki/File:Retina-OCT800.png* in April 2018

- **Optical Frequency Domain Reflectometry (OFDR):** also known as *swept-source SD-OCT*, the source emits narrow-band light at a frequency that varies linearly between a discrete set of values for every A-scan. Each narrow band signal provides higher Signal to Noise Ratio (SNR) with respect to a broadband one [19][18].

The design and development of OCT scanners was mainly driven by the desire of improving the diagnosis. It takes care of tasks previously assigned to biopsy, with the dual advantage of being non-invasive and preserve the samples properties. Although, the great improvements made in OCTs manufacture – specifically in terms of acquisition times – showed the possibility to exploit this technique for real-time monitoring inside the Operating Room (OR); this lead to the development of what it is called **iOCT** [20].

1.2.2   Intraoperative OCT

It took more then a decade for OCT to get its place inside the OR. The first significant drawback to overcome was size [21]. From the advents of technology in medicine, the OR kept becoming more and more busy with devices, with this resulting in the need for any new technology to prove itself a disruptive improvement for both the patients and the operators. When the advantages of having OCT scanners inside the OR became clear, the first efforts were put into increasing its portability. This is the case of some hand-held devices [22; 23; 24; 25; 26] at first, and some mounted devices later on [27; 28; 29; 30; 31; 32].

Initial solutions consisted in real-time B-mode scanners exploiting the new SD-OCT technologies of the early 2000's. The surgeons gained access to cross sectional images; this was a great improvement with respect the previously only available pre- and post-operative monitoring, but yet he/she had to pause the procedure to check the tomographic images. The limit here was that, since their introduction in the first half of the century, clinical microscopes had not been significantly innovated. For this reason the following important step was to integrate the recently available iOCT images with the microscopes already in use. A sample iOCT image is shown in Figure 4.

Leica [33] and Zeiss [34] developed the first two microscope integrated iOCT scanners [35; 36; 37]. Two large research studies tried to assess and validate the objective improvement caused by iOCT introduction: the **PIONEER** study [28; 30] and the **DISCOVER** study [38]. Brought on from 2014 to 2017, the PIONEER study enrolled 531 eyes during the first year, and achieved an overall 98% success rate in image acquisition. It also underlined the great potential

Figure 4: Sample image obtained from a commercially available iOCT scanner. These images are the ones available to the surgeon during a procedure when employing a microscope mounted real-time OCT scanner. The resolution is lower than for the diagnostic OCT, this is compensated by a significantly higher acquisition speed.

for iOCT to influence the surgical decision-making. DISCOVER study, on the other hand, began in 2014 and is still ongoing. It enrolled 1025 eyes, with a planned enrollment amount of 1500. The study achieved a 99% success rate in image acquisition, impacting significantly on the surgical decision-making in 36% of the cases [39].

All this showed the advantages of having such a tool in the OR, but also led to uncover some additional issues and challenges for an effective application of this methodology. Higher computational power, together with more efficient algorithms were necessary to achieve real-

time monitoring without loosing in image quality. Displacements due to surgical maneuvers needed to be somehow compensated, to guarantee reliable imaging. Those kind of problems are the ones being addressed in the following years by the main manufacturing companies [21].

Currently commercially available instrumentation is summed up in Table I based upon information collected in [39]. All the five devices exploits the spectral domain technologies we discussed in 1.2.1, and while *Envisu C2300* is a hand-held scanner, the other four are all microscope integrated devices.

TABLE I: COMMERCIALLY AVAILABLE iOCT SYSTEMS

Manufacturer	Leica Microsystems	Zeiss	Leica Microsystems	Leica Microsystems	Haag-Streit
Product name	Envisu C2300	Rescan 700	EnFocus Ultra-HD OCT	EnFocus Ultra-Deep OCT	Surgical iOCT
FDA approval	2012	2014	2015	2015	2015
Axial res. (μm)	4	5.5	4	9	10(in air)
Imaging depth (mm)	2.5	2	2.5	11.1	4.2(in air)
Scan length (mm)	8–14	3–16	20	20	5–30
Acq. speed ($A\ scans/s$)	32,000	27,000	32,000	18,000	10,000
Wavelength (nm)	870	840	860	880	840

1.2.3   <u>Applicative domain</u>

OCT represented a massive step forward for ophthalmology. This is true for both clinical and academical reasons, since it gave the possibility to get high resolution *in vivo* images of the retina. Given the scope of the project, the most of the focus was on ophthalmic application so far. Although, the features that established its success in ophthalmology were soon identified as potentially problem-solving in other contexts as well [19].

Dermatology often encounters issues which are in some way similar to those of ophthalmology. High resolution volumetric images could only be obtained via direct tissue sampling: a non-invasive technique is clearly a great advantage as it keeps the tissues as they are, besides the significant factor of preventing useless pain for the subject. Furthermore, OCT showed to be a powerful tool for skin related pathologies, such as skin cancer, psoriasis or dermatitis [40].

Odontology took advantage of OCT as well. Soon after the first disclosure of this technique, images of the oral tissue were obtained, which led to useful diagnostic applications [41; 42]. An important contribution, though, was provided in the scope of dental restoration analysis, where a proper monitoring technique was still lacking [43; 44].

Non biomedical applications have been found wherever high-resolution, non-destructive imaging is needed. Precision manufacturing, textile roughness measurements – as well as painting investigation – are examples of industrial applicative domains for this technology [19].

1.3   <u>OCT image processing</u>

As said in section 1.2, OCT was at first developed with diagnostic purposes, as an in-vivo alternative to the common ocular biopsy. Hence, it should not surprise that the main effort

in extracting information from this kind of images was focused on quantitative evaluation of clinically interesting data. Among the others, thickness of the retinal layers is one of the most addressed. The initial approach was the same as for any other kind of medical image analysis, which is manual **segmentation**. It requires time and experience, but once trained the experts are able to identify every interesting layer of the retina, marking interesting features and distinguishing between physiological and pathological conditions.

Segmentation is the process of defining a finite set of homogeneous separate regions inside an image [45]. Manual segmentation performed by experts is always considered the golden standard. This is due to the intrinsic exceptional capacity of the human brain to recognize patterns in 2D images, which makes us incredibly good at performing segmentation. However, computer-aided segmentation could lead to automatic recognition of diseases or particular conditions, saving time and thus increasing the accessibility to monitoring and diagnostics.

Below some of the main segmentation algorithms are described. The subsection **Common algorithms** contains all the classical image processing methods for segmentation, together with some more advanced algorithms. In the **Machine learning** subsection are described all those *expert systems* that can be trained on existing data to perform effective segmentation. A whole subsection is dedicated to the **Watershed** algorithm, given its relevance in the scope of this project. All the algorithms are described for the simplified task of segmenting gray scale images; each method can be implemented in order to deal with 3-channel images.

1.3.1   Common algorithms

The first, intuitive method for segmentation is **thresholding** on a pixel-wise basis. Since the aim is to identify sets of homogeneous pixels, extracting points within a specific range of the gray level histogram is a reasonable way of proceeding. It is simple and computationally inexpensive. Depending on whether the thresholding is performed based on information relative to the whole image, or just to a small location, the algorithm is classified as either *global* or *local* [46; 47]. Local thresholding methods are also referred to as *adaptive* methods. One common limit of those algorithm is that they discard any information related to contiguity between pixel, so additional processing is generally needed to further subdivide the thresholded image into separate and uniform regions.

To overcome this limit, statistics provide us with a whole set of **clustering** algorithm, which work toward the identification of spatially coherent homogeneous groups of pixels. Hoffman and Jain [48] tested some of those methods for image segmentation. Among hierarchical, nearest neighbor, decision trees and squared error clustering, the latter proved to be the most accurate. The advantage of clustering is that it works on features of different nature, and can thus be highly effective. This advantage, though, has the obvious side effect of requiring a feature selection and extraction step. Its robustness is paid with complexity.

A more systematic approach is using filters to perform **edge detection**. Two main types of filters can be identified for this task: sequential and parallel filters. The former class of filters decides if one pixel is part of an edge depending on the classification of other pixels [49]. Parallel filters, on the other hand, depends only on the examined pixel and its close neighbors. The most

common example of parallel filters used for edge detection is that of difference filters. They enhance the rapid changes in gray intensity level by computing a sort of derivative (difference) of the image along one or two directions. First order difference filters perform the actual difference of the pixels, with different sets of weight. This is the case of *Sobel*, *Prewitt* and *Roberts* gradient [50]. Second order filters, such as the *Laplacian*, compute the second derivative. The second derivative goes to zero on linear ramps, so these filters are the best at identifying isolated lines, points or corner edges [51]. Edge detection methods, based on filtering and morphological transformations, have been successfully applied to OCT images; some examples are described in [52; 53]. The main problems related to edge detection techniques based on filtering is that they are generally strongly directional, while an optimal filter should be rotation invariant.

The **relaxation** method is an iterative method. The pixels can be classified in parallel, but at each iteration their classification depends on the result of the previous round [45]. The probability p_{ij} of a pixel q_i being assigned to a specific class C_j is modified according to a measure of compatibility $C(i, j; h, k)$ with the pixel q_h being in a different class C_k and the probability p_{hk} of this pixel to actually belong to the class. In this way a positive compatibility value together with a high probability p_{hk} will increase the probability p_{ij}. A negative or zero value will respectively decrease or leave unmodified the probability.

Another set of iterative methods are based on **Markov Random Fields (MRFs)**, defined as sets of statistical variables that satisfy the Markov local properties [54]. Conceptually similar to Bayesian networks, MRFs are *non-directed* graphs allowing to model cyclic dependencies

among variables. They have been successfully used to resolve efficiently many image processing tasks, such as image restoration and, among the others, image segmentation [55; 56].

1.3.2 Machine learning algorithms

The basic common application of machine learning algorithm is generally either classification or regression. Building upon these two fundamental abilities, many complex systems have been developed to perform tasks that was thought to be exclusively for humans. The fascinating aspect is that these expert systems can learn out of the data, without having to give them any significant a-priori information. In the case of classification – which well fits with segmentation – the general architecture of a model would consist in a feature extractor and a classifier. Generally the first part is where the contextual knowledge is transferred from the developer to the system, while the second is completely trained on experience.

One example of those methods is called **Random Forest Classifier (RFC)**. An array of features is extracted from each image. Those features are then used to train a set of decision trees, and the final classification of a pixel will be an average of the results provided by each tree. To reduce correlation between the trees, features are randomly sub-sampled at each node, and the input to each tree are different subsets of the original dataset [57]. Techniques of this kind have been successfully applied to OCT images segmentation, as well as in other medical contexts [58; 59].

Artificial Neural Networks (ANNs) are a powerful tool, largely deployed in many data analysis applications, particularly for classification task. They can be used to segment images performing pixel-wise decisions. They would require a huge set of data to be trained, but they

proved to be extremely accurate and robust when trained properly [45]. However, finding the right set of features to be extracted from the images is still up to the developer, and requires deep knowledge of both the specific medical context and the general image processing field.

Identifying the proper set of features is actually an issue for both the models just described. The solution is moving toward what is called **deep learning**. The main characteristic of deep learning algorithm is that they can learn both the features and how to classify the data basing upon them. Lee et al. [60] proposed a model to learn the best set of features out of the data. Such a method could be applied to fully automate both RFCs ans ANNs algorithms.

A specific evolution of ANNs, though, has already been proposed and well tested. **Convolutional Neural Network (CNN)** are a class of networks which applies a set of convolution filters on the input data, before feeding the extracted information to an ANN. They have been developed taking inspiration from the human visual cortex peculiar structure [61]. The main advantage of this technique is that both the feature extractor (convolution filters) and the classifier ANN are trainable, so no a-priori knowledge is required to properly process the images.

1.3.3   Watershed

The watershed algorithm is an original and effective method for segmenting images. It was firstly proposed by Vincent and Soile [62] and it has proven to be well suited for image segmentation [63; 64]. The algorithm relies on a very simple concept:

- Consider the 2D image as a 3D surface, where each pixel intensity is converted into a height. Darker regions will be "wells" as lighter regions will be "hills"

- Identify all the wells in the image

Figure 5: Representation of the watershed algorithm via the geographical analogy. The image is considered as a landscape where the brightness is mapped into ground height. The water fills the basins in a simulated flood. The boundaries between adjacent basins (thick black lines) are the contours of the segmented regions.

- Shed water in each well

- Any time the water spills from one well to another, a boundary has been found, and a segment has been defined

Figure 5 shows an intuitive representation of what described above.

The principal documented issue with this method is over-segmentation. As any local minimal can be seen as a well, a noisy image will be segmented into many small areas. In this case a post-processing phase will be necessary in order to merge some of those tiny regions into bigger ones. An interesting technique for avoiding over-segmentation is described in [65].

1.4   Robotic guided surgery

The first documented use of robotic systems in surgery dates back to 1985, with the deployment of the robotic arm Puma 560 (Unimation, Danbury, CT, USA) in a neurosurgical biopsy [66], which is a non-laparoscopic procedure. The first laparoscopic cholecystectomy was performed in 1987. This led to the first FDA authorization of a robotic system for general laparoscopic surgery in 2000, the **da Vinci® Surgery System** (Intuitive Surgical® Inc., Sunnyvale, California). Since then robotic surgery has developed and spread enormously, thanks to its many advantages and the continuous improvement of the deployed technology. Beside the da Vinci, some of the most popular commercially available robotic systems are Naviot (Hitachi Ltd., Tokyo, Japan), RoboDoc (Integrated Surgical Systems, Sacramento, CA, USA) and Zeus and Aesop (Computer Motion, Goleta, CA, USA) [67].

1.4.1   Advantages of robotic surgery

Among all the advantages that caused robotic surgery to gain interest, **movement filtering** is certainly one of the most important. Physiological, small movements of the hands are present even in the most trained and proficient surgeons. Referred to as **tremors**, in some applicative domains this generally is not a problem, as far as the movement range keeps being small with respect to the scale of the region under interest. Robotic assistance offers the possibility to *low-pass* filter the surgeon's maneuvers, canceling all those unwanted, high frequency movements.

Another significant advantage – somehow related to the previous one – is that of **movement scaling**. This allows the surgeon to perform extremely precise operations with a resolution beyond the limits of the human movement control. Also, the limits of the scaling power are

Figure 6: da Vinci SI HD platform (Intuitive Surgical® Inc., Sunnyvale, California)

those of technology, which is in continuous improvement. Those two features, common to the most of the commercially available devices, results in an overall impressive gain in **dexterity**.

Moreover, the continuous exposure of the surgeon to radiation during real-time monitored operations may have harmful consequences in the long term. In this scenario, teleoperated robots can increase the **exposure protection**, as well as giving the possibility to perform **telemedicine**.

1.4.2 Disadvantages of robotic surgery

Beside all the described (1.4.2) strengths, some significant drawbacks limit the use of robotic surgery technologies, leaving room for further improvement. The main one is **cost**. The deployment of robotic systems requires prohibitive expenses for both the purchase and maintenance

of the robot: the da Vinci Surgical System costs around US $1M$, with an annual cost of US $100k$ for the maintenance and services [68].

Additionally, the overall **operating time** is sometimes increased, also due to additional setup time of the OR. There are documented cases [69] in which the same procedure took $20 - 30$ minutes more when performed with the da Vinci then when performed manually. Furthermore, as for any other new technology introduced in the medical field, an initial limit is the **learning curve**.

1.4.3 Robotic surgery in ophthalmology

Robotic surgery has had an incredibly wide and fast spread in many surgical specialties. Neurosurgery and cardiac surgery are some of the main ones, together with urology or nephrology. However, ophthalmic surgery has some particular features that prevented robotic surgery to gain importance at the same pace. Surgeons performing retinal surgery need to train in maneuvering incredibly small tools in en environment where no proprioceptive information is provided, because of the reduced scale. They learn to get information about the relative position between the tools and the eye walls or tissues just from the 2D images given by the microscope or the iOCT. Given the incredible technical difficulties, it would appear normal for robotic surgery to be already assessed in the field. Although, this is not true. The problem with the current technology is mainly related to the high resolution required. Also, all the commercially available robotic systems were designed for a different context, and they do not fit well in ophthalmic applications. Currently there are some ongoing works for the development of robots for applications in ophthalmology [68]. The first in-vivo robot-assisted eye surgery procedure

was performed in 2015 using a **da Vinci SI HD platform**, depicted in Figure 6. The system proved to be appropriate to properly perform the *pterygium*, but no significant improvement was reported [70].

The **Intraocular Robotic Interventional Surgical System (IRISS)** [71] is a robotic platform for performing eye surgery procedures for both the posterior and the anterior segment. It has two manipulators, each one controlled by a joystick. Many different surgical tools can be operated by the manipulators, and it has already been successfully tested in four common, delicate procedures: central vitrectomy with detachment of posterior hyaloids, removal of opaque lens cortex, capsulorhexis and simulation of temporal retinal vein microcannulation [68]. The **John Hopkins Steady-Hand Eye Robot** [72] was developed to share with the surgeon the control of some surgical tools. It allows fine control of the movements, and is provided with a 3 degrees of freedom (DOF) force sensor meant to be used to give audio feedback to the surgeon [73]. A similar result in hand stabilization has been achieved from the **Micron Surgical System** [74]. An amazing negative feedback control system keeps the tip of the instrument steady while the hand is moving. It cancels tremors very efficiently, without the need of any externally mounted support structure.

CHAPTER 2

MATERIALS AND METHODS

The final deliverable of the project is a visuo-haptic simulation of a retinal surgery procedure. The simulation itself is not the main aim of the project though. This work has to be intended as a validation work for the proposed guidance system based on the processing of iOCT images. This chapter gives a detailed description of all the tools exploited to address each specific issue. The main implemented features are described as well, together with some of the technical challenges encountered.

2.1   Image Processing

The guidance provided to the user is constructed entirely on the iOCT B-scans. Therefore, one of the core tasks to take care of was the processing of these images. A specific C++ computer vision library was employed to interface with them and to extract all the relevant information. This library is described below together with the selected dataset and the developed processing algorithm. Since iOCT provides two perpendicular cross-sections at any cycle, the proposed algorithm is applied to two images at a time.

2.1.1   OpenCV

The **Open Source Computer Vision Library (OpenCV)**[75], is a free open-source computer vision library developed in optimized C/C++, which interfaces with C++, Python and Java. Released under a Berkley Source Distribution (BSD) license, it is freely available

for both academic and commercial development. This contributed to its wide dissemination, together with the possibility to take advantage of multi-core execution as well as hardware acceleration (enabled with OpenCL[76]). Its classes and methods provide powerful and simple tools to deal with any computer vision related issue: from basic to advanced image processing, machine learning, object recognition, data visualization and so on.

cv::Mat is the basic class for 2D images. Both single and multi-channel images can be loaded as $n \times m$ matrices of any data type, and specific members and methods allow to access any information about the image (e.g., size, number of channels, etc.) as well as performing basic operations (e.g., cropping, copying, resizing, etc.). Each element of the matrix can be easily accessed and managed exploiting the **cv::Point** class. Each cv::Point object is basically an indexed element $p(i,j)$ with

$$i : i \in \mathbb{N} \ \wedge i < n$$

and

$$j : j \in \mathbb{N} \ \wedge j < m$$

where n and m are respectively the numbers of rows and columns of the image. The methods *cv::imread()* and *cv::imwrite()* can be used to load and save images in any of the most common format, and *cv::imshow()* provides an easy way to show any loaded image, which is very useful when developing custom processing algorithms.

A whole set of built-in filters is also provided. From the basic Median, Gaussian and Moving Average filters to more complex ones, such as *Sobel* for edge detection or many different

thresholding filters. Moreover, each one of these filtering tools has a set of different parameters, allowing its fine tuning to reach the desired result.

The main reason of using OpenCV in the scope of this project was for developing an edge detection algorithm, being both effective and computationally efficient. Many different attempts were performed, deploying smoothing and thresholding techniques together with edge detection filtering methods. The core functionality of the developed algorithm is a well-known segmentation algorithm called **Watershed** [65], accessible via the ad hoc method *cv::watershed()*. A brief introduction to this algorithm can be found in 1.3.3, and the final implementation will be discussed in detail further on.

2.1.2   Data description

In sight of a complete intraoperative guidance system for surgeons performing retinal surgery, the developed project aims to simulate at best a real case scenario. Hence, only information which could be reasonably found in the OR during a procedure could be exploited. This led to decision of using only iOCT images, even if they have lower definition and are generally more noisy than the diagnostic images obtained from a common OCT. An example can be seen in Figure 4.

The chosen images are a set of iOCT scans, representing a volume portion of a human retina. The images properties are summarized in Table II.

The fact that the axial (or depth) resolution is way higher than the lateral (or transverse) resolution is intrinsically related to machine working principle. The axial resolution (see 1.2) can be roughly approximated with the coherence length of light [16], which can be defined as:

TABLE II: iOCT IMAGES SPECIFICATIONS

Nº channels	1
Nº colors	256 ($8bpp$)
Axial resolution (pixel)	1024
Lateral resolution (pixel)	200×200
B-scan spacing	$3.6mm$

$$l_c = \frac{2 \ln 2}{\pi} \times \frac{\bar{\lambda}^2}{\Delta \lambda}$$

Besides the coefficient $(\sim \frac{1}{2})$ the coherence length l_c – hence the axial resolution – is quadratically proportional to the average wavelength $\bar{\lambda}$ and inversely proportional to its spectral width $\Delta \lambda$. In simple words, since the coherence length is generally about one order of magnitude smaller than the average wavelength, the resulting axial resolution will be approximately one order of magnitude bigger than $\bar{\lambda}$ as shown by the following equation:

$$\frac{(a^n)^2}{a^{n-1}} = a^{2n-(n-1)} = a^{n+1}$$

Considering near infrared light ($\bar{\lambda} = 750 \div 1400nm$) yields to reasonable values of μm or tens of μm for the axial resolution.

On the other hand, transverse resolution is limited by a totally different factor: scanning time requirements. This is true only for iOCT, which needs to reach a sufficiently high framerate to provide real-time imaging. Therefore, the solution adopted by the common manufacturer

is that of performing **sparse A-scan** acquisition, rather than full raster scans. Of course having images with axial resolution 2-4 times better than the lateral one will result in distorted images. Hence, iOCT visualization systems have to *stretch* the obtained image and fill the missing parts with interpolation techniques. The original images have been modified off-line, under the assumption that in a real-case scenario the incoming images would be already re-sampled.

The set of images taken into consideration, as said before, represents a volume. This means that stacking them one on top of the other the volume can be reconstructed. If an image can be considered as a 2D matrix, a volume can easily be seen as a 3D matrix, whose sections are the original images. Each pixel is thus extruded into a voxel, filling the space between consecutive slices. It is clear, under those assumptions, that the resolution of the obtained volume will depend on the distance between each successive B-scan as well as on their lateral resolution.

2.1.3   Image segmentation

The process designed to extract valuable information from those images is summarized below:

1. **Template matching**

   A template image was created from a sample of retinal boundary region. This squared sample can be swiped across the image as a convolution filter. Convolution is a common operation in signal (or image) processing. It consists in substituting a value with the weighted sum of some surrounding points. The power of this technique consists in its ability to keep into consideration local relationships between points, as well as its invariance to rigid transformations. Convolution filters are an effective way to spot significant

features in images, regardless of their position or orientation. The resulting output will be a smaller image, where the higher intensity areas will be those resembling the most the template.

2. **Blurring**

The images are corrupted by a high frequency noise, which affects significantly the identification of the edges. Also –looking for just one border, separating two macro areas in the image– over segmentation might be a significant issue. Given that the template matching procedure enhances the interesting regions, blurring the obtained image is a good way to ignore non interesting areas, thus reducing the risk of spotting more boundaries than needed. The blurring is performed by a simple low-pass linear filter, which averages the values of the neighbor pixels.

3. **Markers identification**

Identifying the markers is a fundamental step of the algorithm, and the reason lies behind the nature of the segmentation algorithm described in the next point. Markers are roughly segmented regions selected a priori, necessary in order to attempt a finer segmentation only in interesting zones. The first step to identify those markers is binarization via thresholding. This is the best way to separate foreground and background. Among the various types of thresholding, the selected one is called **Otsu's Method**, after its inventor [77]. Briefly, it consists in selecting the optimal threshold by minimizing the inter-class variance between the two sets of colors (black and white). Additionally, some trials revealed that inverting the colors of the binarized image brought to better result.

Subsequently a morphological operation called **opening** is applied. This has the effect of sub-estimating the black area. At this point the markers will be all the remaining black regions.

4. **Segmentation**

All the steps described above were necessary to properly accomplish this last point. The **watershed** algorithm (introduced in 1.3.3) takes the name from its working principle. Some regions of the image are identified as wells: in our case, the markers identified at the previous step. The reason why they are referred to as wells (or *basins*) is that this method considers the gray scale image as a 3D landscape, where white is the high ground. This landscape is progressively filled with "water", as if it was flooded. As the level of water increases some of the basins will merge, meaning that a boundary between two adjacent region has been found. Following the analogy, the algorithm will found all the *dims* separating the basins, and will store them as an array of pixels belonging to the original image.

5. **Surface interpolation**

Once all the different regions (or segments) are identified, the retinal surface will be identified by the outermost boundary points: this is ensured by the anatomy of the eye. In order to reconstruct a smooth and continuous surface, an interpolation method was used. First of all the set of points is filtered: limits on the local derivative and on the distance between the average of the others points avoid the use of outliers in the surface

reconstruction. Then, a Catmull Rom spline [78] is used to create a line interpolating a subset of the available points.

6. **Averaging**

Filtering the points before the interpolation is usually not enough to keep the surface reconstruction smooth. Hence, an additional averaging step is necessary. At each iteration the reconstructed surface is stored to be averaged with the successive ones. Up to three previous surfaces are used to smooth each new interpolation. The best result of this technique are obtained when the successive frames represents portions that are close in space, so anatomically similar.

A visual representation of the described algorithm is provided in Figure 7. The watershed algorithm is a simple albeit powerful tool, but its efficacy relies entirely on the processing phase that precedes it. In particular, this algorithm is known to be sensitive to the problem of over-segmentation. An accurate selection of the markers is the only way to prevent it. The template matching allows to enhance regions of the image which are usually not well separated from the surrounding areas. This gives the opportunity to isolate only the interesting bounding line, which is that of the retina. As can be noted in Figure 7, sometimes the epiretinal membrane might be spotted as if it was a retinal contour. Forcing the proximity between successive points of the boundary helps in avoiding this kind of misclassification.

2.2 Virtual environment

The virtual simulation is a Windows console application developed in the **Microsoft Visual Studio** environment. The whole program was written in C++, deploying a set of computer

Figure 7: Work flow of the implemented processing algorithm. The red arrows represent respectively: template matching, low pass filtering, thresholding, morphological opening, watershed segmentation, contours extraction.

graphics libraries to build the interactive 3D environment. Below are described those libraries and how they have been used to integrate the information extracted from the iOCT images (see 2.1) and the hardware device used to provide the actual guidance.

2.2.1   <u>LACE Library</u>

LACE Library is a C++ software library developed at University of Illinois at Chicago (UIC) during the development of works described in [79; 80; 81; 82]. It merges four free libraries into one powerful tool for software development in the scope of **Virtual Reality (VR) and Haptics** console applications. It is provided as a Microsoft Visual Studio 10 project. Its structure is depicted in Figure 8, and it includes the following libraries:

- **QuickHaptics™ (QH):** micro API of Geomagic OpenHaptics® Toolkit, which provides a simple programming interface for haptic devices [83].

- **Wykoby Computational Geometry Library© (WK):** open-source multi-platform C++ library for computational geometry in 2D/3D environment [84].

- **Visualization Library (VL):** open-source C++ library based on OpenGL 4 to manage 2D/3D graphics rendering [85].

- **Ascension (AS):** library used to interface with electromagnetic tracking sensors, provided by *Ascension* Technology Corporation© [86].

Each one of the four libraries can be activated and used separately, and the two main Application Programming Interfaces (APIs), QH and VL, work on separate threads, taking care of the haptic and visual rendering respectively. Any object in the graphics domain of VL

Figure 8: Schematic representation of LACE Library structure. Four libraries communicate with different environments and share the information exploiting the specific implementation of LACE.

may or may not have its haptic counterpart in the QH environment and vice-versa. This is an important feature for what concerns performances, since the collision detection algorithm implemented in QH – which is the core algorithm of the library – is computationally heavy. The amount, size and complexity of the haptically rendered objects impact significantly the overall speed of the program, which in particular affects the reliability of the tactile feedback: it is provided by a servo loop which needs to work at approximately $1000Hz$ to be effective.

The complexity of LACE arises from the necessity to merge information coming from different environments. It exploits the effective visual rendering classes of VL while allowing the control of a haptic device, which is working on a separate thread and on a completely different set of informations. It allows to include information coming from the external world employing the electromagnetic trackers, and to use all the gathered data in the WK environment to ease and enhance the computation. This complexity, though, requires a smartly organized structure so as to guarantee an effective interface to the user. This led to the definition of four main groups of classes, each one somehow in charge of communicating with a specific environment and sharing the acquired information with the others.

These groups are:

- **Renderable objects classes:** the classes implementing all the graphically and haptically renderable objects.

- **Rendering classes:** the classes allowing to deal with multiple renderings in one application; the *LACE_CuttingPlane*, to see the internal structure of any graphically rendered object, belongs to this group as well.

- **Tracking system classes:** classes taking care of the communication with the electro-magnetic trackers (Ascension Technology Corporation$^{©}$)

- **Force classes:** classes implementing the specific force effects for some classes of render-able objects, namely LACE_Volume and LACE_Extrusion.

All the classes belonging to the set of renderable objects, inherit from the base class *LACE_Object*. This class implements the communication between the two environments of VL and QH. The integration of all the sets of classes described above is performed by the *LACE_Class*. It contains a set of pointers to the other classes and takes charge of their initialization and update during the execution of the program. A schematic representation is provided in Figure 9.

Besides the fact that it provides a simple interface for programming haptic devices, LACE Library was taken in consideration for the development of this project because of its significant focus on medical data visualization: a whole class, named **LACE_Volume**, has been implemented to take care of volume rendering from medical images. It allows to load and visualize volumes out of **.mhd* files and the associated raw data, providing the possibility to apply intensity transfer functions, just like common volume visualization software (e.g., VolView[87], ParaView[88]). Though medical volumes are not supported by QH default classes, a custom haptic effect has been implemented for those volumes, exploiting QH servo loop callbacks. The class **LACE_VolumeForce** – a member of LACE_Volume – allows to *touch* those volumes, whose tactile properties are related to each voxel intensity level. Being dealt with as a 3D matrix of voxels, each volume has got its own coordinate system based on voxels position. A

Figure 9: Schematic representation of the LACE_Class structure. Four main groups of classes are identified: Renderable objects classes, Force classes, Tracking system classes and Rendering classes. They implement communication and integration with the classes of the four libraries constituting LACE.

specific voxel can be identified either with a set of 3 coordinates or with a single index. Ad hoc method of the LACE_Volume class allows to obtain and exploit these values.

2.2.2 Volume rendering

The created scene aims at reproducing a real situation, taking into account all the obvious limitations related to the virtual environment. Rather than an exact representation of reality, the project wants to create a reliable validation of the proposed method using only information

that would be available in a real-case scenario. The virtual scene created has the only purpose of easing the user interaction.

The selected dataset, described in 2.1.2, was used to reconstruct a volumetric representation of a portion of the retina. The images are provided with some information about the machine used to collect them, which was fundamental to build the volume with the right proportion. In particular lateral, vertical and depth spacing were the most important value to consider. The dataset consists in a set of 200 *bmp* B-scans, corresponding to a square of $6 \times 6mm$ on the surface of the retina. The images were loaded in Matlab (The MathWorks Inc., Natick, MA, USA) [89] and used to construct a 3D matrix containing the cube data. Each voxel was then converted from integer (int) to an unsigned char (uchar). This causes a loss in the overall information but reduces the space required in memory, thus increasing the performance. Information loss is not a major problem, since the real computation for the guidance is performed on the original images. Besides the ease in dealing with images and matrices, Matlab was selected for this task due to its unique interface with medical data. The Medical Images Processing Toolbox offers many useful functions to deal with medical data such as DICOM or, among the others, *.mhd files (see 2.2.1). In particular, the function *write_mhd()* creates an *.mhd file with the respective *.raw file, which contains the data stored in the 3D matrix. To get an effective visualization of the volume, two steps were necessary. The images, as said before, are very noisy. An impulsive noise is spread all over the image, and this reflects in a bad reconstructed volume. Hence, the first step was to smoothen the original images with a median filter. This kind of filter was selected because it acts specifically against the impulsive

Figure 10: Sample of the implemented scene. The reconstructed volume (on the left) is arbitrarily rotated to improve visualization. The cursor (black dot) is interacting with it by manipulating a haptic device. On the right, the two perpendicular cross-sections show the internal structure of the retina. The cursor is mapped in those images and is displayed as a red dot. The red lines lying on the interfaces are the detected edges. The point right beneath the cursor will be used to compute the distance cursor/surface. Some lateral points will be used to compute the normal to the surface.

noise (see 2.1.3). The following step was to enhance the visibility of the interesting region of the volume. To do so, the created volume was opened in VolView [87]. This software allows to apply transfer functions to the volume to isolate different tissues depending on their voxel intensity. In this case, the light reflectivity of each point. The actual volume files (*.mhd and *.raw) are not modified by this operation, but the transfer function values can be stored and applied on the volume once it is loaded in the application.

The volume can be loaded as a LACE_Volume object. There are two ways to render volumes with LACE library. This depends on how they are managed by VL. The **ray casting** rendering mode allows to apply many transfer functions. However, rendering a volume with the ray casting mode does not provide an actual volumetric representation, it generates a 2D image from volumetric data. Graphically the volume looks right, but interacting with it in the 3D space is like interacting with a planar figure. The alternative is the **slice based** rendering mode. This modality is simpler, it basically builds the volume by stacking and interpolating each *slice*. The possible transfer functions are limited, but the intensity channel can be tuned accessing the VL_Actor member of the LACE_Volume object. A lower and a upper threshold, tuned with VolView, can be used to generate a linear transfer function and select the interesting regions. The constructed volume can be seen in Figure 10, together with the cross-sectional images.

An additional pre-processing step was performed to ease the three-dimensional perception. The intensity values of the volume were rescaled, proportionally to their height. This transformation was applied only to those voxels with intensity level overcoming a certain threshold, in order to allow the use of the transfer function described before to remove the background. Mapping the intensity level with the height allowed to apply different colors at each height level.

2.2.3   Cursor mapping

Since the information is extracted by the section images, but the user interacts with the reconstructed volume, there was the need to have consistency between the two spaces. To do

so, the device is tracked inside the volume by checking the coordinates of its closest voxel. The 3D coordinates inside the volume contains information about which section the user is interacting with (Z coordinate) and where inside this section the cursor is exactly located (X and Y coordinate). The volume was created with the same images that are now being processed, but some scaling factors had to be considered. The images we are using, as said in 2.1.2 had to be re-sized to compensate for the low lateral resolution of the scanning device. The volume, however, is created with the original images. Its proportions are correct because of the proper spacing factor between each slice. Therefore, a scale factor equal to the slice spacing value had to be applied to the cursor position to map it into the images.

To improve the usability of the simulation it was necessary to allow the user to rotate and translate the volume in order to adjust its point of view. The procedure just described has the great advantage that it exploits the volume own reference system, which will follow any transformation applied to the volume. An intuitive representation of this procedure is proposed in Figure 11. However, moving the volume leads to different problems. The cursor has to be mapped inside an image only if it is actually inside the volume. When the volume is aligned with the axes of the space coordinate system this is a trivial task. When the volume dimensions are known, a simple control on the cursor position is enough to establish if it is inside or outside that region. Moreover, VL provides a member for any VL_Actor which is called Axis Aligned Bounding Box (AABB). The AABB, intuitively, is a box aligned with the axis which perfectly delimits the volume. It comes useful due to its method *AABB::isInside()* which takes a position as only argument and returns a boolean value determining if the position belongs to

Figure 11: Schematic representation of the cursor mapping procedure. The coordinates of the cursor inside the volume (left) are used to identify the same point inside the image (right). A scaling factor is needed to match the two sets of coordinates. This method is insensitive to rotations or translations applied to the volume, since the latter has its own reference system. The x coordinate inside the volume defines which cross-section the point needs to be mapped in.

the bounding box. The problem with this built in solution is that a AABB, as suggested by the name, keeps being aligned with the axes of the static reference system. When the volume is rotated the region of space it occupies does not change, but the bounding box, constrained to being axis aligned, needs to modify its dimension so as to cover contain the volume in its new orientation. Of course this will cause the whole method to fail.

The adopted solution is conceptually similar. A virtual bounding box is constructed around the volume when it is still aligned with the axes. Whenever it is rotated or translated, the transformation function is stored, so as to keep track of where each voxel is with respect to its

initial position. To control if the cursor is inside the bounding box of the volume, we need to get the inverse matrix of the volume transformation and apply it to the cursor position. At this point, if the *virtual* position of the cursor is inside the initial bounding box, the real cursor will be in the real bounding box.

As specified in 2.1 two perpendicular cross-sections are considered at any time frame. Hence, the mapping method described above is analogously applied to both of them.

2.3   Haptic feedback

The final objective of all what have been described so far is to build and provide to the user an effective and reliable guidance during the operation. To accomplish this a haptic device was deployed and controlled based on the information extracted from images by the application. The next sections describe in detail the employed device and how the ad hoc force feedback was designed and implemented.

2.3.1   Haptic device

As stated before, the project consists in a visuo-haptic simulation. The haptic feedback is given to the user via the **Touch™ 3D Stylus** (3D Systems®, Rock Hill, SC, USA). It is a 3-DOF robotic arm working on a $1kHz$ servo loop, capable of providing a reliable force feedback. Some specifications about the utilized device can be found in Table III, as provided by the manufacturer [83].

There are two macro categories of haptic devices: **admittance** devices and **impedance** devices. The first class refers to devices which senses the externally applied force to compute and generate a consistent velocity. Impedance type haptic devices do the exact opposite. They

TABLE III: TECHNICAL SPECIFICATIONS OF THE TOUCH™ 3D STYLUS

Force DOF	3
Position DOF	6
Potentiometers linearity	$\pm 5\%$
USB	2.0
Force feedback workspace ($W \times H \times D$)	$10.45 \times 9.5 \times 3.5$" ($26.54 \times 24.13 \times 8.89 cm$)
Height (arm at rest)	7.0" ($17.78 cm$)
Base diameter	5.5" ($13.97 cm$)
Nominal position resolution	$0.084 mm$
Max. force	$3.4N$

sense a velocity by means of encoders placed in the joints and output an appropriate force. Those terms come from an analogy with the electrical circuits: mechanical impedance is defined as the ratio between the force and the velocity (voltage over current), and admittance is the inverse, velocity over force [90]. Due to this difference, the two kinds of devices are suitable for different applications. Admittance type devices simulate a static reaction according perfectly with the Coulomb friction model, since they can output a zero velocity from any input force. This is not possible for impedance devices, being impossible to sense the zero velocity when any force is applied. However, impedance devices work perfectly in a dynamic situation, whereas admittance devices can only approximate the exact behaviour [91].

The device previously introduced, pictured in Figure 12, belongs to the class of impedance type devices. It is a low-cost device which fits well in simulations that do not need an extremely

Figure 12: Touch™ 3D Stylus (3D Systems®, Rock Hill, SC, USA)

high level of accuracy in the provided tactile sensation. It interfaces with the QH microAPI described in 2.2.1, which makes it fast and easy to develop complex force effects in interaction with a 3D virtual environment.

2.3.2 Force feedback

The general idea behind the guidance system is to prevent the surgeon to damage the retina while getting in close proximity with it, for instance during vitrectomies or epiretinal membrane peelings. Hence, the main information extracted from the iOCT images to be exploited is the distance between the operating tool and the retinal surface. Once both the position of the cursor and the position of the retina right behind it are known (see 2.1.3), computing the distance is a trivial task. In the process of building an effective force effect to guide the surgeon, this distance has two fundamental roles. First of all, it can be continuously monitored and compared to a user defined threshold. Once this threshold is overcome, the force effect is activated and

the distance value can be used to defined the instantaneous desired magnitude of the force to provide to the user.The designed effect can be simply seen as a one dimensional spring, whose behaviour is described by the simplified **Hooke's law**:

$$F = k \times \Delta x$$

The force F is proportional to the displacement of the spring from a reference position, by means of an elastic constant – or gain. In our case, the reference position is the distance threshold. Any movement from this position toward the retina acts as compressing the spring,. So for us

$$\Delta x = d_t - d_c$$

where d_t is the distance threshold and d_c is the current distance. This yields to a linear relation between the distance and the applied force, which basically increases the resistance felt by the user when approaching the retina.

Deciding the magnitude of the force is just the first step. Tho have an effective feedback, the right direction needs to be defined as well. An initial method could be to generate a vertical force, pointing away from the retina. The problem with this rough solution is that the image is discrete. Hence, when moving parallel to the retinal surface each different position detected will be a discrete step, causing the device to "jump". To recreate the sense of touch, as if we were actually touching the retina, we need to compute the normals to the surface we are looking

Figure 13: Schematic representation of the developed procedure to compute the normal forces. The magnitude is proportional to he displacement Δx with respect to the threshold distance d_t. The direction is perpendicular to the line connecting the two red points, which are symmetric with respect to the detected edge (green point).

at. The method developed to solved this is quite simple, and is represented schematically in Figure 13.

The segmentation algorithm gives us an irregular line which represents the detected surface, and to compute the distance we are using just one of the points composing this line. If we consider also two points symmetric with respect to the central one, and we connect them with a straight line, this line is a good approximation of the surface. The direction perpendicular to the line, so, is approximately the direction of the normal. This is actually a normal of the

planar section we are looking at. If we do the same thing on both the perpendicular cross sections we will obtain two lines, which indeed identify a plane. The normal to this plane can be easily computed as the vectorial sum of the two planar normals, shown in Figure 14.

The same complication described for the cursor mapping in 2.2.3 arises here as well when having to deal with the volume rotation. In fact, the normals are computed based just on the cross section images, which do not rotate with the volume. However, if the volume is rotated we want the force feedback to be consistent with what we see in the virtual scene. To solve this the approach was similar to the one adopted in 2.2.3. Instead of the creation of a virtual box, this time was created a reference system, a group of three perpendicular vectors parallel to the scene principal axes. Any transformation applied to the volume during the execution of the program is applied to this custom reference system as well. Thus, this reference system is in agreement with the volume own reference system. All what remains to do, is converting the components of the normals computed from the images to coordinates of this new reference system.

Having to deal with a delicate task, it was necessary to take some precautions for what concerns the impulsive forces that may result from the spring-like model, in particular having to deal with noisy images. This precaution was a simple filter of the force magnitude, which is averaged with the force applied to the previous frame. Additionally, a safety *cap* was imposed to the total applied force, which cannot overcome a certain fixed value.

Figure 14: Schematic representation of the procedure to compound the normal forces computed on two perpendicular planes (ZX and ZY) into one 3D normal force. The two planar normal force are derived as explained in Figure 13, and the resulting force is the sum of the two forces. The result is then normalized to avoid unwanted scaling factors.

CHAPTER 3

DISCUSSION

The aim of this project was to develop a guidance system for retinal surgery procedures. The guidance was designed based on information extracted from iOCT images. This work has to be intended as an initial stage of a long term project. A simulated environment was developed to allow usability and primary evaluation of the algorithms and the haptic guidance implementation. The core functionality is the image processing algorithm, which to be effective should be able to extract information from the scans in real-time during the operation. The concept of real-time though is intrinsically fuzzy, since it depends on the context. Briefly, it means to be able to catch up with the pace at which incoming information is provided. In our case, this lower limit is given by iOCT scanning frequency.

The dataset used in this work is made of images with a resolution of 1024×200 pixels. This means that each B-scan is a raster of 200 A-scans. Considering Table I (in 1.2.2), the maximum acquisition speed for the commercially available iOCT scanners is $32.000 A - scan/sec$. This yields to a theoretical maximum acquisition frequency of

$$f_B = \frac{32.000 \ Ascan/sec}{200 \ Ascan/Bscan} = 160 \ Bscan/sec = 160 Hz$$

which means a time-span of around $6.25 ms$ for taking care of the image processing and the guidance construction. This frequency is actually higher than the real scanning rates

Figure 15: Right: normalized distribution of the execution time of the image segmentation algorithm presented in 2.1.3(left). Left: box plot of the same distribution enhancing visualization of the median (red line) and the outliers (red crosses)

commonly reported by the manufacturers. However, it acts as a conservative threshold for evaluating the feasibility of the implemented algorithm. In addition to the theoretical scanning frequency, in fact, the reconstruction of the image and the transmission of the data cause a lag, which reflects into a lower frame-rate. The execution time of the implemented processing algorithm (described in 2.1.3) have been recorded 2234 times. Graphs in Figure 15 display the distribution of the obtained results. The maximum execution time recorded is of $5ms$, yielding to a theoretical processing frequency of $5ms^{-1} = 200Hz$ as worst-case-scenario. This value would be enough to satisfy the requirements, however the time recorded is just the time needed for the edge-detection only. A few millisecond margin to be considered for taking into account

TABLE IV: TECHNICAL SPECIFICATIONS OF THE TESTING PC

CPU model	AMD FX(tm)-4300 Quad-Core Processor
CPU clock frequency	$3.80GHz$
RAM	$8GB$
OS	64-bit MS Windows 10 Home
GPU model	NVIDIA GeForce GT 730
GPU memory clock	$1.8Gbps$

the data storage and the creation of the force effect. The average time required for executing the algorithm, though, is of $2.9ms \pm 8\%$, which corresponds to a processing frequency of $\sim 344Hz$, reasonably enough for a real-time processing. As shown by the box plot in Figure 15, this value is affected by a significant amount of outliers on the upper portion. The specifications of the machine used to test the algorithms are summed up in Table IV.

An additional result of this work is a contribution to the development of LACE Library (see 2.2.1). The extensive use of this API inevitably required the debugging of some utility implementations. This should help to increase the reliability of the library and ease its employment for future developer and users. Moreover, some of the classes which were initially conceived for highly specific tasks were modified in order to improve their generalization capabilities. The proposed work hence gives an additional validation of the deployment of LACE Library in the domain of visuo-haptic applications development.

Finally, the integration of LACE Library with OpenCV has proven to be feasible and effective, which opens the way to a possible inclusion of this opensource API to LACE Library. This would enrich it with powerful tools for dealing with image processing and computer vision.

CHAPTER 4

CONCLUSIONS

A primary implementation of a novel real-time guidance system for retinal surgery is proposed. iOCT is already largely used in the OR as a support in performing delicate intraocular procedures. This project shows a way to exploit the information provided by these non-invasive imaging devices to enhance the range of perception of the surgeon. While the sub-micrometer scale of the maneuvers is too fine compared to the physiological resolution of the human somatosensory system, the proposed method allows to recreate a reliable tactile sensation.

The developed implementation has shown the potential of this method. iOCT images were effectively processed, and a simple tactile feedback were created based upon the information extracted from them. Acquisitions of the processing time required for a single image reported that the developed algorithm is theoretically feasible in terms of data throughput, to be integrated with commercially available real-time scanners, without limiting their performance. However, precise evaluation of the segmentation accuracy is still missing. For this reason future works will aim at compare the implemented algorithm with a human performed segmentation as ground truth.

Other future developments might work in the direction of overcoming some of the main limitation of the project as it is now. First of all, the haptic device currently used is far from what could be actually used in a real-case retinal surgery operation. Both the spatial resolution and the intrinsic friction are too high to be effectively employed in a surgical procedure. Higher

quality, fully transparent haptic devices could be exploited to reduce the friction and have a better resolution on the control force.

Having a finite dataset rather than an actual time series of images clearly prevents the application to reproduce a real-like situation. Having more images, also, would make an evaluation of the algorithm more significative. Eventually this could also allow to exploit machine learning algorithm, which are generally more robust and fast.

The lack of the possibility to interface with a real OCT machine is another limitation. This could allow to measure the actual time window available for processing images, providing a more reliable evaluation of the over all developed system. It could also give the chance to explore more efficient ways to implement the communication between the scanning device and the processing system, as well as potentially smarter methods to process the incoming data for noise removal or feature extraction.

The current implementation relies entirely on the interpretation of the images performed by the processing system. This is in accordance with the common direction of making devices smart and independent. However, a guidance system that integrates its own methods with the real experience of the operator would definitely be more reliable. It is reasonable to believe that this would positively affect the level of trust of both the surgeon and the patient, increasing the chances of actually improving the performance.

Retinal surgery is still far from relying on robotic surgery as it happened in other medical fields. Many procedural and technological limits concur in slowing this process. A hybrid system as the one presented in this work could be a further step in getting close to that integration,

as it supports the surgeon enhancing his/her grasp of the status of the ongoing procedure. Additional value of an actual integration of this system with a real iOCT machine would have the significant advantage of exploiting only already existing technology.

APPENDIX

Copyright Clearance Center **RightsLink®**

[Home] [Account Info] [Help] [✉]

ELSEVIER

Title:	Optical Coherence Tomography: An Emerging Technology for Biomedical Imaging and Optical Biopsy
Author:	James G. Fujimoto,Costas Pitris,Stephen A. Boppart,Mark E. Brezinski
Publication:	Neoplasia
Publisher:	Elsevier
Date:	January–April 2000

Copyright © 2000 Neoplasia Press, Inc. Published by Elsevier Inc.

Order Completed

Thank you for your order.

This Agreement between University of Illinois at Chicago -- Matteo De Silvestri ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

printable details

License Number	4330941181479
License date	Apr 16, 2018
Licensed Content Publisher	Elsevier
Licensed Content Publication	Neoplasia
Licensed Content Title	Optical Coherence Tomography: An Emerging Technology for Biomedical Imaging and Optical Biopsy
Licensed Content Author	James G. Fujimoto,Costas Pitris,Stephen A. Boppart,Mark E. Brezinski
Licensed Content Date	January–April 2000
Licensed Content Volume	2
Licensed Content Issue	1-2
Licensed Content Pages	17
Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	3
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Original figure numbers	figure 1,2,3
Title of your thesis/dissertation	Real-time Haptic Guidance System for Retinal Suregry based on Intraoperative Optical Coherence Tomography
Expected completion date	May 2018
Estimated size (number of pages)	90
Attachment	
Requestor Location	University of Illinois at Chicago 1200 WEST HARRISON STREET CHICAGO, IL 60607 United States Attn: University of Illinois at Chicago
Publisher Tax ID	98-0397604
Total	0.00 USD

[ORDER MORE] [CLOSE WINDOW]

CITED LITERATURE

1. Kwok, A. K., Lai, T. Y., and Yuen, K. S.: Epiretinal membrane surgery with or without internal limiting membrane peeling. Clinical & experimental ophthalmology, 33(4):379–385, 2005.

2. Haimann, M. H., Burton, T. C., Brown, C. K., et al.: Epidemiology of retinal detachment. Arch Ophthalmol, 100(2):289–292, 1982.

3. Huang, D., Swanson, E., Lin, C., Schuman, J., Stinson, W., Chang, W., Hee, M., Flotte, T., Gregory, K., Puliafito, C., and et, a.: Optical coherence tomography. Science, 254(5035):1178–1181, 1991.

4. Kent, C.: Intraoperative oct coming into focus. 2014. Online; Accessed: 2018-04-10.

5. Stephenson, M.: Advancements in surgical microscopes. May 2015. Online; Accessed: 2018-04-13.

6. Butler, R.: The anatomy of the compound eye ofperiplaneta americana l. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology, 83(3):223–238, 1973.

7. Johnston, P.: Traumatic retinal detachment. British journal of ophthalmology, 75(1):18–21, 1991.

8. Fisher, S. K. and Lewis, G. P.: Müller cell and neuronal remodeling in retinal detachment and reattachment and their potential consequences for visual recovery: a review and reconsideration of recent data. Vision research, 43(8):887–897, 2003.

9. Epiretinal membranes. https://www.asrs.org/patients/retinal-diseases/19/epiretinal-membranes. Online; Accessed: 2018-04-15.

10. De Bustros, S., Thompson, J. T., Michels, R. G., Rice, T. A., and Glaser, B. M.: Vitrectomy for idiopathic epiretinal membranes causing macular pucker. British Journal of Ophthalmology, 72(9):692–695, 1988.

CITED LITERATURE (continued)

11. Sadaka, A. and Giuliari, G. P.: Proliferative vitreoretinopathy: current and emerging treatments. Clinical ophthalmology (Auckland, NZ), 6:1325, 2012.

12. Hilton, G., Machemer, R., Michels, R., Okun, E., Schepens, C., and Schwartz, A.: The classification of retinal detachment with proliferative vitreoretinopathy. Ophthalmology, 90(2):121–125, 1983.

13. Machemer, R., m Aaberg, T., Freeman, H. M., Alexander, R. I., John, S. L., and Ronald, M. M.: An updated classification of retinal detachment with proliferative vitreoretinopathy. American journal of ophthalmology, 112(2):159–165, 1991.

14. Fujimoto, J., Pitris, C., Boppart, S., and Brezinski, M.: Optical coherence tomography: An emerging technology for biomedical imaging and optical biopsy. Neoplasia, 2(1-2):9–25, 2000.

15. Grasselli, J.: On the relative motion of the earth and the luminiferous ether. Applied Spectroscopy, 41(6):933–935, 1987.

16. Fercher, A. F., Drexler, W., Hitzenberger, C. K., and Lasser, T.: Optical coherence tomography - principles and applications. Reports on Progress in Physics, 66(2):239, 2003.

17. Walsh, A.: Spectral domain oct: An a to z guide. The Journal of Ophthalmic Photography, 30(1):10–15, 2008.

18. Yaqoob, Z., Wu, J., Yang, C., et al.: Spectral domain optical coherence tomography: a better oct imaging strategy. Biotechniques, 39(6), 2005.

19. Freitas, A., Amaral, M., and Raele, M.: Optical coherence tomography: development and applications. In Laser pulse phenomena and applications. InTech, 2010.

20. Carrasco-Zevallos, O. M., Viehland, C., Keller, B., Draelos, M., Kuo, A. N., Toth, C. A., and Izatt, J. A.: Review of intraoperative optical coherence tomography: technology and applications. Biomedical optics express, 8(3):1607–1637, 2017.

21. Ehlers, J.: Intraoperative optical coherence tomography: past, present, and future. Eye, 30(2):193, 2016.

22. Chavala, S., Farsiu, S., Maldonado, R., Wallace, D., Freedman, S., and Toth, C.: Insights into advanced retinopathy of prematurity using handheld spectral domain optical coherence tomography imaging. Ophthalmology, 116(12):2448–2456, 2009.

23. Dayani, P., Maldonado, R., Farsiu, S., and Toth, C.: Intraoperative use of handheld spectral domain optical coherence tomography imaging in macular surgery. Retina, 29(10):1457–1468, 2009.

24. JP, E., Kernstine, K., Farsiu, S., Sarin, N., Maldonado, R., and Toth, C.: Analysis of pars plana vitrectomy for optic pit-related maculopathy with intraoperative optical coherence tomography: a possible connection with the vitreous cavity. Arch Ophthalmol, 129(11):1483–1486, 2011.

25. PB, K., Kaufmann, C., Menke, M., Watson, S., and Bosch, M.: Use of intraoperative fourier-domain anterior segment optical coherence tomography during descemet stripping endothelial keratoplasty. Am J Ophthalmol, 150(3):360–365, 2010.

26. Scott, A., Farsiu, S., Enyedi, L., Wallace, D., and Toth, C.: Imaging the infant retina with a hand-held spectral-domain optical coherence tomography device. Am J Ophthalmol, 147(2):364–373, 2009.

27. Ray, R., Baranano, D., Fortun, J., Schwent, B., Cribbs, B., and Bergstrom, C. e. a.: Intraoperative microscope-mounted spectral domain optical coherence tomography for evaluation of retinal anatomy during macular surgery. Ophthalmology, 118(11):2212–2217, 2011.

28. Ehlers, J., Dupps, W., Kaiser, P., Goshe, J., Singh, R., and Petkovsek, D. e. a.: The prospective intraoperative and perioperative ophthalmic imaging with optical coherence tomography (pioneer) study: 2-year results. Am J Ophthalmol, 158(5):999–1007, 2014.

29. Ehlers, J., Grifth, J., and Srivastava, S.: Intraoperative optical coherence tomography during vitreoretinal surgery for densevitreous hemorrhagein the pioneer study. Retina, 35:2537–2542, 2015.

30. Ehlers, J., Petkovsek, D., Yuan, A., Singh, R., and Srivastava, S.: Intrasurgical assessment of subretinal tpa injection for submacular hemorrhage in the pioneer study utilizing intraoperative oct. Ophthalmic Surg Lasers Imaging Retina, 46(3):327–332, 2015.

31. Ehlers, J., Tam, T., Kaiser, P., Martin, D., Smith, G., and Srivastava, S.: Utility of intraoperative optical coherence tomography during vitrectomy surgery for vitreomacular traction syndrome. Retina, 34(7):1341–1346, 2014.

32. Ehlers, J., Xu, D., Kaiser, P., Singh, R., and Srivastava, S.: Intrasurgical dynamics of macular hole surgery: an assessment of surgery-induced ultrastructural alterations with intraoperative optical coherence tomography. Retina, 34(2):213–221, 2014.

33. Leica microsystems. Online; Accessed: 2018-04-12.

34. Zeiss meditec. Online; Accessed: 2018-04-12.

35. Ehlers, J. P., Tao, Y. K., Farsiu, S., Maldonado, R., Izatt, J. A., and Toth, C. A.: Integration of a spectral domain optical coherence tomography system into a surgical microscope for intraoperative imaging. Investigative ophthalmology & visual science, 52(6):3153–3159, 2011.

36. Tao, Y. K., Ehlers, J. P., Toth, C. A., and Izatt, J. A.: Intraoperative spectral domain optical coherence tomography for vitreoretinal surgery. Optics letters, 35(20):3315–3317, 2010.

37. Binder, S., Falkner-Radler, C. I., Hauger, C., Matz, H., and Glittenberg, C.: Feasibility of intrasurgical spectral-domain optical coherence tomography. Retina, 31(7):1332–1336, 2011.

38. Ehlers, J. P., Kaiser, P. K., and Srivastava, S. K.: Intraoperative optical coherence tomography using the rescan 700: preliminary results from the discover study. British journal of Ophthalmology, pages bjophthalmol–2014, 2014.

39. Uchida, A., Srivastava, S. K., and Ehlers, J. P.: Update on the intraoperative oct: Where do we stand? Current Ophthalmology Reports, 6(1):24–35, Mar 2018.

40. Mogensen, M., Thrane, L., Jørgensen, T. M., Andersen, P. E., and Jemec, G. B.: Oct imaging of skin cancer and other dermatological diseases. Journal of biophotonics, 2(6-7):442–451, 2009.

41. Feldchtein, F. I., Gelikonov, G., Gelikonov, V., Iksanov, R., Kuranov, R., Sergeev, A. M., Gladkova, N., Ourutina, M., Warren, J., and Reitze, D.: In vivo oct imaging of hard and soft tissue of the oral cavity. Optics express, 3(6):239–250, 1998.

42. Colston, B. W., Everett, M. J., Da Silva, L. B., Otis, L. L., Stroeve, P., and Nathel, H.: Imaging of hard-and soft-tissue structure in the oral cavity by optical coherence tomography. Applied Optics, 37(16):3582–3585, 1998.

43. WANG, X.-J., MILNER, T. E., DE BOER, J. F., ZHANG, Y., PASHLEY, D. H., and NELSON, J. S.: Characterization of dentin and enamel by use of optical coherence tomography. SPIE milestone series, 165:586–590, 2001.

44. Fried, D., Glena, R. E., Featherstone, J. D., and Seka, W.: Nature of light scattering in dental enamel and dentin at visible and near-infrared wavelengths. Applied optics, 34(7):1278–1285, 1995.

45. Pal, N. R. and Pal, S. K.: A review on image segmentation techniques. Pattern recognition, 26(9):1277–1294, 1993.

46. Hall, E.: Computer image processing and recognition. Elsevier, 1979.

47. Gonzalez, R. C. and Woods, R. E.: Digital image processing, 2012.

48. Hoffman, R. and Jain, A. K.: Segmentation and classification of range images. IEEE transactions on pattern analysis and machine intelligence, (5):608–620, 1987.

49. Davis, L. S.: A survey of edge detection techniques. Computer graphics and image processing, 4(3):248–270, 1975.

50. Rosenfeld, A.: Digital picture processing. Academic press, 1976.

51. Hall, E.: Computer image processing and recognition. Elsevier, 1979.

52. Moraes, M. C., Cardenas, D. A. C., and Furuie, S. S.: Automatic ioct lumen segmentation using wavelet and mathematical morphology. In Computing in Cardiology (CinC), 2012, pages 545–548. IEEE, 2012.

53. Moraes, M. C., Cardenas, D. A. C., and Furuie, S. S.: Automatic stent segmentation in ioct images using combined feature extraction techniques and mathematical morphology. In Computing in Cardiology Conference (CinC), 2013, pages 1215–1218. IEEE, 2013.

54. Blake, A., Kohli, P., and Rother, C.: Markov random fields for vision and image processing. Mit Press, 2011.

55. Hansen, F. and Elliott, H.: Image segmentation using simple markov field models. Computer Graphics and Image Processing, 20(2):101–132, 1982.

56. Derin, H., Elliott, H., Cristi, R., and Geman, D.: Bayes smoothing algorithms for segmentation of binary images modeled by markov random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, (6):707–720, 1984.

57. Lang, A., Carass, A., Swingle, E. K., Al-Louzi, O., Bhargava, P., Saidha, S., Ying, H. S., Calabresi, P. A., and Prince, J. L.: Automatic segmentation of microcystic macular edema in oct. Biomedical optics express, 6(1):155–169, 2015.

58. Lang, A., Carass, A., Hauser, M., Sotirchos, E. S., Calabresi, P. A., Ying, H. S., and Prince, J. L.: Retinal layer segmentation of macular oct images using boundary classification. Biomedical optics express, 4(7):1133–1152, 2013.

59. Antony, B. J., Abràmoff, M. D., Harper, M. M., Jeong, W., Sohn, E. H., Kwon, Y. H., Kardon, R., and Garvin, M. K.: A combined machine-learning and graph-based framework for the segmentation of retinal surfaces in sd-oct volumes. Biomedical optics express, 4(12):2712–2728, 2013.

60. Lee, S. H., Koo, H. I., and Cho, N. I.: Image segmentation algorithms based on the machine learning of features. Pattern Recognition Letters, 31(14):2325–2336, 2010.

61. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

62. Vincent, L. and Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis & Machine Intelligence, (6):583–598, 1991.

63. Beucher, S.: Use of watersheds in contour detection. In Proceedings of the International Workshop on Image Processing. CCETT, 1979.

64. Meyer, F. and Beucher, S.: Morphological segmentation. Journal of visual communication and image representation, 1(1):21–46, 1990.

65. Bieniecki, W.: Oversegmentation avoidance in watershed-based algorithms for color images. In Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2004. Proceedings of the International Conference, pages 169–172. IEEE, 2004.

66. Kwoh, Y.: A new computerized tomographic aided robotic stereotactic system. Robot Age 7, pages 17–21, 1985.

67. Cleary, K. and Nguyen, C.: State of the art in surgical robotics: clinical applications and technology challenges. Computer Aided Surgery, 6(6):312–328, 2001.

68. Nuzzi, R. and Brusasco, L.: State of the art of robotic surgery related to vision: brain and eye applications of newly available devices. Eye and brain, 10:13, 2018.

69. Salman, M., Bell, T., Martin, J., Bhuva, K., Grim, R., and Ahuja, V.: Use, cost, complications, and mortality of robotic versus nonrobotic general surgery procedures based on a nationwide database. The American surgeon, 79(6):553–560, 2013.

70. Bourcier, T., Chammas, J., Becmeur, P.-H., Danan, J., Sauer, A., Gaucher, D., Liverneaux, P., and Mutter, D.: Robotically assisted pterygium surgery: first human case. Cornea, 34(10):1329–1330, 2015.

71. Rahimy, E., Wilson, J., Tsao, T., Schwartz, S., and Hubschman, J.: Robot-assisted intraocular surgery: development of the iriss and feasibility studies in an animal model. Eye, 27(8):972, 2013.

72. Gonenc, B., Handa, J., Gehlbach, P., Taylor, R. H., and Iordachita, I.: A comparative study for robot assisted vitreoretinal surgery: Micron vs. the steady-hand robot. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 4832–4837. IEEE, 2013.

73. Üneri, A., Balicki, M. A., Handa, J., Gehlbach, P., Taylor, R. H., and Iordachita, I.: New steady-hand eye robot with micro-force sensing for vitreoretinal surgery. In Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on, pages 814–819. IEEE, 2010.

74. MacLachlan, R. A., Becker, B. C., Tabarés, J. C., Podnar, G. W., Lobes Jr, L. A., and Riviere, C. N.: Micron: an actively stabilized handheld tool for microsurgery. IEEE Transactions on Robotics, 28(1):195–212, 2012.

75. Bradski, G.: The opencv library (2000). Dr. Dobbs Journal of Software Tools, 2000.

76. Stone, J. E., Gohara, D., and Shi, G.: Opencl: A parallel programming standard for heterogeneous computing systems. Computing in science & engineering, 12(3):66–73, 2010.

CITED LITERATURE (continued)

77. Zhu, N., Wang, G., Yang, G., and Dai, W.: A fast 2d otsu thresholding algorithm based on improved histogram. In Pattern Recognition, 2009. CCPR 2009. Chinese Conference on, pages 1–5. IEEE, 2009.

78. Twigg, C.: Catmull-rom splines. Computer, 41(6):4–6, 2003.

79. TAGLIABUE, E.: Visuo-haptic model of prostate cancer based on magnetic resonance elastography: a feasibility study. 2017.

80. Gatti, C.: Evaluation of haptic virtual fixtures in psychomotor skill development for robotic surgical training. 2017.

81. Rapetti, L., Crivellaro, S., De Momi, E., Ferrigno, G., Niederberger, C., and Luciano, C.: Virtual reality navigation system for prostate biopsy. In Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, page 35. ACM, 2017.

82. Faso, A.: Haptic and Virtual Reality Surgical Simulator for Training in Percutaneous Renal Access. Doctoral dissertation, 2017.

83. Geomagic®: Quickhaptics™ microAPI. http://www.geomagic.com/it/products/open-haptics/overview. Online; Accessed: 2018-04-09.

84. Partow, A.: Wykobi computational geometry library. http://www.wykobi.com/index.html, 2005. Online; Accessed: 2018-04-09.

85. Bosi, M.: Visualization library. http://www.visualizationlibrary.org/, 2011. Online; Accessed: 2018-04-09.

86. Ascension technology corporation©. https://www.ascension-tech.com/products/. Online; Accessed: 2018-04-09.

87. Kitware, I.: Volview. https://www.kitware.com/volview/. Online; Accessed: 2018-04-09.

88. Ahrens, J., Geveci, B., and Law, C.: Paraview: An end-user tool for large data visualization. Visualization Handbook, 2005.

89. The mathworks, inc. Online; Accessed: 2018-04-18.

CITED LITERATURE (continued)

90. Mihelj, M. and Podobnik, J.: Haptics for virtual reality and teleoperation, volume 67. Springer Science & Business Media, 2012.

91. Stewart, D. and Trinkle, J. C.: An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, volume 1, pages 162–169. IEEE, 2000.

VITA

NAME	Matteo De Silvestri

EDUCATION

Spring 2017 - On date	Master of Science in Bioengineering and Biomedical Engineering, University of Illinois at Chicago, USA
Fall 2016 - On date	Master Degree in Biomedical Engineering – Technologies for Electronics (BTE), Politecnico di Milano, Milan, Italy
Fall 2013 - fall 2016	Bachelor's Degree in Biomedical Engineering (BE), Sep 2016, Politecnico di Milano, Milan, Italy

LANGUAGE SKILLS

Italian	Native speaker
English	Full working proficiency
	2017 - TOEFL examination (102/120)
	A.Y. 2017/18 One Year of study abroad in Chicago, Illinois
	A.Y. 2017/18. Lessons and exams attended exclusively in English
Spanish	Elementary proficiency

SCHOLARSHIPS

N/A	N/A

TECHNICAL SKILLS

Advanced level	Problem Solving
Advanced level	Signal Processing
Advanced level	Image Processing
Advanced level	Augmented Reality
Advanced level	VR and Haptics
Advanced level	Programming (C, C++, Matlab)
Good level	Programming (Python, Lua, Processing)

WORK EXPERIENCE

VITA (continued)

Jan 2018 - May 2018	Graduate Research Assistant, UIC Department of Ophthalmology and Visual Sciences, Chicago, Illinois

PUBLICATIONS

spring 2016	A. Solazzo, E. Del Sozzo, I. De Rose, M. De Silvestri, G. C. Durelli, M. D. Santambrogio. Hardware Design Automation of Convolutional Neural Networks. *2016 IEEE Computer Society Annual Symposium on VLSI, IEEE*

PROJECTS

Ongoing	**Real-time Guidance System for Retinal Surgery based on Intraoperative Optical Coherence Tomography**. Development of a Visual C++ application to provide real-time guidance the surgeon during vitreo-retinal surgery procedures. Guidance is based on the analysis of intraoperative OCT images, exploiting OpenCV library functionalities.
Fall 2017	**Emotion recognition**. EEG data from a publicly available dataset were used to train Support Vector Machine (SVM) classifiers. The aim was to recognize emotions by correctly placing them within a two dimensional system (arousal vs. valence). Possible uses of this method can be found in Brain Machine Interface (BMI) applications.
Spring 2017	**Dance Dance Revolution, for fingers!** Development of a tabletop version of the famous game. An Arduino board was used to process the data, and carbon black sheets were used to implement the capacitive touch sensors. The graphic user interface was developed in Processing, and the whole work was documented in a final report.
Summer 2016	**Eye Tracker**. The aim of the project is to develop a digital writing system based on eye tracking. A first prototype has been implemented, employing a webcam for the image acquisition, an FPGA-equipped board for the data processing, and a monitor showing the virtual keyboard.
Spring 2016	**Hardware accelerated Convolutional Neural Networks for the automation of medical images analysis**. An FPGA based acceleration of Convolutional Neural Networks (CNNs) is proposed as a tool for the analysis of diagnostic images. More in particular the aim of the project is to implement such a tool for the identification of cancerous tissue in mammographic images, training the network on a public dataset.

VITA (continued)

Spring 2016   **CNN$_2$ECST**. CNN$_2$ECST is a web automation tool for the synthesis of the hardware implementation of Convolutional Nerual Networks (CNNs). Starting from a high level description of a trained network, the framework returns the scripts needed to implement the system on the desired FPGA-based device.