WIA-DM

Ophthalmologic Decision Support System based on Clinical Workflow and Data Mining Techniques

Sistema de Apoio à Decisão para a especialidade de Oftalmologia baseado em técnicas de Workflow e Data Mining aplicadas em dados obtidos em Ambiente Clínico

Project Report
Version 1.0

Edgar da Silva Ferreira – Nr. 501022514

July 2, 2007

Centro Cirúrgico de Coimbra

Physics Department
FACULTY OF SCIENCES AND TECHNOLOGY
UNIVERSITY OF COIMBRA

Document Revisions

Date	Version	Description	Author
2007-06-14	0.1	Document creation	Edgar Ferreira
2007-06-26	0.2	Document updated to ISA template	Edgar Ferreira
2007-06-29	0.3	Document updated after supervisors revision	Edgar Ferreira
2007-07-02	1.0	Final version	Edgar Ferreira

Acknowledgments

After this long journey I must thank to those who made critical contributions on both scientific and personal points of view.

To Doctor António Travassos, project mentor, for giving me the opportunity to develop this ambitious project.

To Engineer Jorge Saraiva, project supervisor, for showing me how an idea can be developed to become a product.

To Eng. José Basílio Simões, Eng. Jorge Landeck and Eng. Lara Osório for such a positive attendance at ISA.

To Dr. Rui Proença, Eng. Robert van Velze, Nurse Odete Videira and Mr. Nuno Videira for their patience and cooperation.

To Prof. Carlos Correia and Prof. Luís Requicha Ferreira for the multidisciplinary contribution they give to this project.

To project partners, Armanda and Paulo, for supporting me and providing a pleasant environment.

To my parents and sister for accepting my absence and giving me courage.

To my girlfriend, Sónia, for her extreme patience and support given at all times.

To everyone who gave their positive contribution to this project.

Abstract

Nowadays one can see a wide application of computer systems in Medicine, especially in data acquisition, registry and supply, constituting the so called Electronic Patient Record (EPR) which allows providing information for advanced consultation and processing. The work described in this report is an integrating part of a project that aims the creation of an innovator system for medical diagnosis support in Ophthalmology, in cooperation with a notable clinic. This system makes use of Data Mining techniques on intelligent analysis and processing of medical data in order to improve all stages of clinical process and to allow a better resources management. In the course of this project, methods for system integration were researched and developed on a modular software structure that is established on a database for RCE storing. Examination results are imported from medical equipments and saved on this database, which provides structured data to patient management module. Developed software solution already presents the required vigour for testing on clinical environment. Based on spatial and snake filtering, algorithms for processing retinal tomography images were created in order to detect both retinal anterior and epithelium pigment limits. The high success rates reached on image processing allowed integrating it in developed software, as well as getting a vision of Data Mining module.

Keywords: Electronic Patient Record, medical diagnosis, Ophtalmology, Data Mining, C-Sharp, Matlab, snake algorithm

Resumo

Actualmente, a utilização de meios informáticos na Medicina, sobretudo ao nível da aquisição, registo e transmissão de dados que constituem o chamado Registo Clínico Electrónico (RCE) é cada vez mais frequente, uma vez que permite disponibilizar a informação para consulta e processamento avançados. O trabalho descrito neste relatório está integrado num projecto que visa a criação de um sistema inovador de apoio ao diagnóstico médico na área da Oftalmologia, em estreita colaboração com uma clínica de referência nesta área. Este sistema utiliza técnicas de Data Mining no tratamento e processamento inteligente dos dados médicos de modo a optimizar as etapas do processo clínico, o que permite por sua vez maior rentabilização de recursos. No decorrer do projecto foram estudados e criados mecanismos de integração de sistemas recorrendo a uma estrutura de software modular, centrada numa base de dados dedicada ao armazenamento do RCE. Os resultados dos exames médicos são importados a partir dos respectivos equipamentos e armazenados nesta base de dados que disponibiliza a informação de modo organizado a um módulo de gestão de pacientes e consultas. A solução de software já desenvolvida apresenta a robustez necessária para o início de testes em ambiente clínico. Foram criados algoritmos de processamento de imagens tomográficas da retina, baseados em filtragem espacial e métodos de snakes, com o objectivo de detectar o limite anterior da retina e o limite do epitélio pigmentado. As elevadas taxas de sucesso atingidas no processamento de imagem permitiram a integração deste elemento no software desenvolvido, bem como a projecção do módulo de Data Mining para um futuro próximo.

Palavras-chave: *Registo Clínico Electrónico, diagnóstico médico, Oftalmologia, Data Mining, C-Sharp, Matlab, algoritmos de snakes*

Contents

Document Revisions, I
Acknowledgments, II
Abstract, III
Resumo, IV
Index of Tables, VII
Index of Diagrams, VII
Index of Figures, VIII
Definitions and Acronyms, IX

1. Introduction 1
1.1. Domain 2
1.2. Objectives 2
1.3. Document Structure 3

2. Project Management 4
2.1. Project Team 4
2.2. Project Coordination 4
2.2.1. Tasks Partition 4
2.2.2. Supervising at CCC 5
2.2.3. Supervising at ISA 5
2.2.4. Initial Planning 5
2.2.5. Final Planning 6

3. State of the Art 7
3.1. Study at CCC 7
3.1.1. Software 7
3.1.2. Examination Equipments 7
3.2. Search on Data-Mining 8

4. Clinical Process Study 10
4.1. Entities Relationship 10
4.2. Workflow 11
4.2.1. Model 11
4.2.2. Persons 12

5. Eye Diseases Study 14
5.1. Overview 14
5.1.1. Anatomy and Diseases 14
5.1.2. Equipment Features 15
5.1.3. Study Issues 16
5.2. Retinopathies 16
5.3. OCT Detectable Diseases 20

6. Software Development 22
6.1. Architecture 22
6.2. Requisites and Prototype 23
6.3. Implementation 25
6.3.1. Main Interface 25
6.3.2. OCT-SLO Acquisition Module 29
6.4. Challenges 35
6.5. Software Tests 36

7. Image Processing 38
7.1. Requisites 38
7.2. Implementation 39
7.2.1. Image Cut 39
7.2.2. Retinal Pigment Epithelium Detection 41
7.2.3. Retinal Anterior Limit Detection 46
7.3. Algorithm Tests 51
7.4. Software Integration 52

8. Conclusions and Future Purposes 54
8.1. Conclusions 54
8.2. Future Purposes 55

References 57

Appendices 58

Index of Tables

Table 1 - Project Team 4
Table 2 - Project Main Tasks Schedule 6
Table 3 - Technical Features of Examination Equipments 8
Table 4 - Entities Interactions in Clinical Process 11
Table 5 - Events Interactions in Clinical Process 11
Table 6 - Workflow tasks of several entities 13
Table 7 - Main Eye Diseases 15
Table 8 - Analysis of Examination Equipments Features 16
Table 9 - Retinopathy Characteristics 19
Table 10 - Significant Parameters in OCT Images 21
Table 11 - Plugin Interface Properties 26
Table 12 - MI Initialization Features 27
Table 13 - MI Load Module Features 27
Table 14 - MI Utilities 28
Table 15 - AM Initialization Features 30
Table 16 - AM Examination Requests Features 31
Table 17 - AM Image Selection Features 32
Table 18 - AM Examination Report Features 33
Table 19 - Software Development Challenges 35
Table 20 - Software Tests 37
Table 21 - Spatial Filtering Stages on RPE Detection 41
Table 22 - Spatial Filtering Stages on RAL Detection 47
Table 23 - RPE Detection Tests 52
Table 24 - RAL Detection Tests 52
Table 25 - Image Processing Features Implemented on Software Module 52
Table 26 - February Time Table 58
Table 27 - March Time Table 58
Table 28 - April Time Table 59
Table 29 - May Time Table 59
Table 30 - June Time Table 59

Index of Diagrams

Diagram 1 - Gantt Diagram of Project Tasks 6
Diagram 2 - Hierarchical Relation in Clinical Process 10
Diagram 3 - Main Workflow Model 12
Diagram 4 - Doctor Workflow Model 12
Diagram 5 - Clinical Chain of Study Issues 16
Diagram 6 - Software Architecture 22
Diagram 7 - Main Interface Requisites 24
Diagram 8 - Acquisition Module Requisites 24
Diagram 9 - Image Processing Requisites 39

Index of Figures

Figure 1 - Human Eye 14
Figure 2 - Acquisition Module Prototype 25
Figure 3 - Main Interface Overview 26
Figure 4 - Plugin Interface Properties 26
Figure 5 - MI Login Window 27
Figure 6 - Administrator Module Layout 28
Figure 7 - MI Change Password Window 29
Figure 8 - AM Initial Layout 29
Figure 9 - AM Waiting Dialog 31
Figure 10 - AM Image Selection Tab Page 32
Figure 11 - AM Examination Requests Window 33
Figure 12 - AM Examination Report Tab Page 34
Figure 13 - AM Image Preview Window 34
Figure 14 - OCT-SLO Acquired Images 39
Figure 15 - OCT Acquired Image 40
Figure 16 - Image Cut Stages. 40
Figure 17 - Spatial Filtering Results on RPE Detection 42
Figure 18 - Starting Points Achieved by Horizontal Scan on RPE Detection 43
Figure 19 - Starting Point Selection by Vertical Scanning on RPE Detection 44
Figure 20 - Illustration of a Left to Right Snake Scanning on RPE Detection 45
Figure 21 - RPE Detection on an OCT-SLO Image 46
Figure 22 - Spatial Filtering Results on RAL Detection 47
Figure 23 - Left and Right Starting Points on RAL Detection 48
Figure 24 - Starting Point Selection by Vertical Scanning on RAL Detection 49
Figure 25 - Retinal Hole Detection 50
Figure 26 - RAL Detection on an OCT-SLO Image 51
Figure 27 - Image Procesing Layout on Patient Management Module 53

Definitions and Acronyms

Acronym	Description
AM	OCT-SLO Acquisition Module
AMD	Age-related Macular Degeneration
DB	Database
CCC	Centro Cirúrgico de Coimbra
C#	C-Sharp
DICOM	Digital Imaging and Communications in Medicine
DM	Data Mining
EPR	Electronic Patient Record
FCA	Free and Clarified Assent
GUI	Graphical User Interface
ID	Identification Number
ISA	Intelligent Sensing Anywhere, Lda.
MI	Main Interface
OCT	Optical Coherence Tomography
RAL	Retinal Anterior Limit
RPE	Retinal Pigment Epithelium
WIA-DM	Workflow & Artificial Intelligence – Data Mining (*reduced term of the project*)

1. Introduction

The use of informatics extends to several aspects of our life. In fact, computer science already plays an important role in Medicine, especially in acquisition and sorting of medical generated data. This evolution allows material and human resources optimization, thus improving the quality of provided services.

However, health professional performance is still surrounded by a great fraction of inherent subjectivity caused by doctor knowledge and experience, or even his current lucidity.

The evolution of computer science in order to become a real support to doctors, joining human knowledge and computing power, seems like the next logical step to make.

Several technologic solutions have been introduced to health care sector through last years. From these tools Electronic Patient Record (EPR) stands out due to its wide implementation. With EPR it is intended to register text data (traditionally hand wrote) as well as to integrate a set of diagnosis multimedia resources (e.g. images, videos, ECG and EEG data, laboratory results and even administrative data). Thus, therapeutic data become structured and available in order to allow an advanced processing.

Data Mining is an analytic process designed to explore large amounts of data in search of consistent patterns and systematic relationships between variables. The process of data mining consists of three stages: the initial exploration, model building or pattern identification, and deployment (i.e., the application of the model to new data in order to generate predictions) [1]. As a multidisciplinary area Data Mining is captivating researchers' interest on several domains [2]:

- Automatic Learning and Artificial Intelligence
- Pattern Recognition from inferred knowledge
- Artificial Neural Networks
- Databases and Data Warehouse
- Statistics and Mathematics
- Data Visualization

Workflow can be seen as a Data Mining consequence because pattern detection from large amounts of data intends to improve the systematic organization of resources. Thus, after knowledge extraction and sorting it is possible to improve information flows, into a work process that can be documented and learned.

Clinical workflow is a subjective process that depends on involved entities and environmental conditions. The alliance of data mining techniques with an objective view of clinical process may significantly increase productivity.

1.1. Domain

WIA-DM is a project integrated on Biomedical Engineering Course of the Faculty of Sciences and Technology of the University of Coimbra. It was started at August 28, 2006 and its ending matches the delivering date of this document: July 2, 2007.

Main corporate entities involved on this project are CCC (Centro Cirúrgico de Coimbra) and ISA (Intelligent Sensing Anywhere).

1.2. Objectives

The main purpose of this project is to create an innovator system that provides helpful tools for medical diagnosis support and resources management. Studying clinical methods in both diagnostic and treatment phases will allow an intervention in these areas, so as putting together acquired know-how and computing power in order to optimize the whole process.

The three main challenges of this project are:

- System integration
- Diagnostic support
- Clinical workflow support

System integration consists on creating interconnection mechanisms between doctors and examination equipments. Those equipments should store results on a server that provides all the data to health professionals, on a structured and accessible way. In this area, the solution passes to create a dedicated database that is supported by equipment-link, patient, consultation and examination management software. It is also a target to make the link between clinical and administrative areas which are, at the time, managed by closed solutions that cause tasks duplication, like patients registry. Thus, one of main proposed challenges is to organize dispersed patient data into a structured EPR.

On diagnostic support unit the object is to create algorithms for detection of characteristics on examination images and data. Thus, the process that nowadays depends exclusively on doctor analysis should be automated, while complementary data, like dimensions or statistics, are provided.

Clinical workflow support can be described as the context analysis stage, which means that acquired knowledge and experience is used to evaluate the variables that take part on medical decision process, and thus to propose the most convenient solutions from the point of view of service and health professionals productivity. The challenge is to use intelligent computing techniques (e.g. data-mining) that could be able to replicate human knowledge.

1.3. Document Structure

This document is divided in eight chapters. In the first chapter (Introduction) a brief thematic introduction is made and main project objectives are pointed.

In the second chapter (Project Management) the project team is described as well as initial and final planning.

In the third chapter (State of the Art) is presented de study of CCC software conditions as well as some literature on data mining methods.

In the fourth chapter (Clinical Process Study) hierarchical and workflow relations at CCC are described.

In the fifth chapter (Eye Diseases Study) biological principles of eye diseases are described. Important diseases for project course are deeply described.

In the sixth chapter (Software Development) software requisites, implementation and tests are described.

In the seventh chapter (Image Processing) developed algorithms for image features detection are described.

In the eighth chapter (Conclusions and Future Purposes) project issues are analysed and conclusions are described. A future vision of the project is proposed.

2. Project Management

This section describes the project structure, emphasizing the work developed by me, Edgar Ferreira, during all stages. This chapter explains project scheduling in five main aspects which include tasks partition, the supervising role of involved companies and both initial and final schedules.

2.1. Project Team

This project was developed by students Armanda Santos, Edgar Ferreira and Paulo Barbeiro, and was supervised by Eng. Jorge Saraiva at *ISA – Intelligent Sensing Anywhere* and Dr. António Travassos at *CCC – Centro Cirúrgico de Coimbra*.

Several persons gave their positive contribute to this project, as it is shown on Table 1.

Name	Contribute	Contact
Armanda Santos	Student	santos.amn@gmail.com
Edgar Ferreira	Student	edgar.bajouca@gmail.com
Paulo Barbeiro	Student	paulobarbeiro@gmail.com
Eng. Jorge Saraiva	Supervisor	jsaraiva@isa.pt
Dr. António Travassos	Supervisor	info@ccci.pt
Dr. Rui Proença	Clinical Collaborator	+351 239 802 705
Eng. Lara Osório	Engineering Collaborator	losorio@isa.pt
Prof. José Basílio Simões	Engineering Collaborator	jbasilio@isa.pt
Prof. Carlos Correia	Engineering Collaborator	correia@lei.fis.uc.pt
Prof. Luís Requicha Ferreira	Engineering Collaborator	requicha@ci.uc.pt
Prof. Jorge Landeck	Engineering Collaborator	jlandeck@isa.pt

Table 1 - Project Team

2.2. Project Coordination

2.2.1. Tasks Partition

During all project stages, differenced targets were defined for involved students. This report describes the tasks that I developed:

- Study of CCC clinical process
- Study of eye diseases, with incidence on retinopathies
- Creation of the software module for OCT-SLO equipment integration
- Creation and improvement of algorithms for feature extraction on OCT-SLO images

These tasks, combined with tasks developed by remaining project students, do not allow getting a final market product. However, created basis and predicted plans for the near future allow, at this time, the definition of the products that will be developed and sold by *BlueWorks, Medical Expert Diagnosis, Lda*, a company founded in the course of this project. This company is shared by companies, doctors, teachers and students who join the project.

2.2.2. Supervising at CCC

The supervision at CCC clinic was guaranteed by Doctor António Travassos, the major mentor of this project. Doctor Rui Proença gave an important collaboration to the project by teaching important subjects on eye diseases as well as Engineer Robert van Velze, who described examination equipments and clinical software systems that exist at CCC. Nurse Odete Videira cooperated on clinical process study and Mr. Nuno Videira cooperated on CCC administrative process study.

Meetings that defined the foundation of *BlueWorks* took place at CCC.

2.2.3. Supervising at ISA

Engineers Jorge Saraiva and Lara Osório provided the supervision at ISA enterprise. Professor Basílio Simões (enterprise COE) also gave his contribution for the project course.

Software development and image processing phases took place at ISA and all enterprise collaborators gave total cooperation to this project.

2.2.4. Initial Planning

This is an ambitious project that seeks the creation of an innovator product; therefore, it needs a deadline longer then the duration of the subject (nine months).

The project schedule was defined during its progress because a significant part of tasks (e.g. ophthalmologic knowledge acquisition) could not be constrained to an inflexible time period. There was an initial idea of important project tasks but a rigorous time table could not be previously defined.

2.2.5. Final Planning

Table 2 shows main tasks description and duration. Detailed schedule of software development and image processing stages are shown on Appendix 1.

Index	Task	Duration	Start	Finish
1	Study of the State of Art	8 days	28-08-2006	06-09-2006
2	CCC clinical process analysis	11 days	07-09-2006	21-09-2006
3	Study of OCT detectable parameters	23 days	27-09-2006	27-10-2006
4	Deep study of retinal diseases	31 days	30-10-2006	11-12-2006
5	*Exam period*	40 days	12-12-2006	05-02-2007
6	Creation of OCT-SLO Acquisition Module	12 days	06-02-2007	21-02-2007
7	1st Intercalary presentation	2 days	22-02-2007	23-02-2007
8	Implementation of advanced functionalities	32 days	26-02-2007	10-04-2007
9	Test and optimization of OCT-SLO Acquisition and Main Interface Modules	15 days	11-04-2007	01-05-2007
10	Image processing	13 days	02-05-2007	18-05-2007
11	Test and optimization of image processing algorithms	6 days	21-05-2007	28-05-2007
12	2nd Intercalary presentation	4 days	29-05-2007	01-06-2007
13	Software and Image processing documentation	8 days	04-06-2007	13-06-2007
14	Project Report	13 days	14-06-2007	02-07-2007

Table 2 - Project Main Tasks Schedule

Diagram 1 illustrates the weekly chronogram of tasks developed during project course.

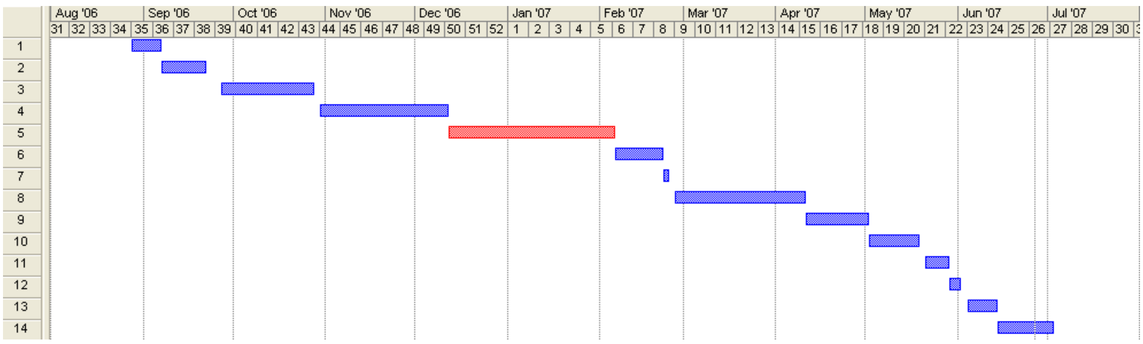


Diagram 1 - Gantt Diagram of Project Tasks

3. State of the Art

The state of the art was analyzed according to two major points of view: CCC working methods with software and equipments, and search for image processing and data-mining based solutions.

This study was not much extensive because it was realized, right from the start, that solutions for this project must be suitable to specific reality, instead of being based on existing solutions.

3.1. Study at CCC

3.1.1. Software

Before searching on CCC suitable integration techniques a study on main features of existing computer systems was performed. Thus, three operating types of software can be described:

- CMOC: this application is installed on CCC consultation sector server and provides consultation management to few consultation rooms. This system allows seeing examination images that are stored on predefined folders. Image filenames are defined by hand on a recognizable manner.
- HIGIA: this is the administrative application that serves CCC internment sector. This system allows registering patient consumptions as well as inventorying financial operations.
- Examination equipments software: examination equipments are controlled by proprietary applications. As described on next section, interconnection possibilities are different for each device.

This search proved that CCC computer systems cannot be directly connected to each other because clinical and administrative systems are provided by different manufacturers, where as examination equipments are controlled by closed solutions.

3.1.2. Examination Equipments

A study of existing CCC examination equipments was performed. Examination devices are external to consultation rooms and are available for examination prescription, by internal and external doctors.

Nine equipments were analysed in order to describe the three main features that are required in system integration. Table 3 shows studied technical features.

Equipment	Operating System	Network Connection	Output
Angiography Retinography	Windows 2000	Connected	Printer Screenshots
GDx™ Scanning Laser System	-	Not connected	Printer
Specular Microscopy	Windows 95	Not connected	Printer
PEC (Perimetry)	Owner OS	Not connected	Printer
Micro Perimetry	Windows 95	Not connected	Printer
VIS (Visumetry)	Windows 95	Not connected	Printer
Stratus OCT (Zeiss)	Windows 2000	Connected	Printer Screenshots
OCT-SLO (OTI)	Windows XP	Connected	Printer Screenshots
Orbscan	Windows 2000	Connected	Printer Screenshots

Table 3 - Technical Features of Examination Equipments

In order to store examination images from CMOC connected equipments, the use of *Snagit*, is necessary as it is a tool that makes screenshots and saves images on a predefined server path. User must specify a filename that identifies current patient.

For not connected equipments the only available output is a paper print or, in some cases, a floppy disk drive.

3.2. Search on Data-Mining

As it was previously related, a clinical integration solution that satisfies all project requisites does not exist. Thus, searching of knowledge was based on techniques that could be applied to developed system.

Data Mining

There is not a defined set of methods that can characterize data mining process. However the meaning of data mining (please refer to Introduction chapter) allows to mention Antonio Valerio Netto doctorate thesis [3].Netto describes the use of data and image analysis techniques in order to get diagnosis on ocular refraction. Some described methods are:

- Machine learning techniques
- Gabor wavelet transform
- Artificial Neural Networks
- Support Vector Machines

Image Processing

An essential part of this project is image processing, namely retinal OCT image processing. Some developed methods were searched on this field.

One of searched resources is Fernández's article [4] in which texture analysis and complex diffusion filtering techniques are applied. However, the use of these methods is limited to image shape, which means algorithms work properly on regular retinal images but they are not tested on images presenting deformations (diseases). In addition, feature detection rates are not demonstrated.

Another searched resource is Luís-Garcia's article [5]. This article demonstrates the use of adaptive snake algorithms on radiograph features detection. Snake algorithms for edge detection are applied on several image processing stages of this project. However algorithms must be fine-tuned to the type of processed images.

Neural Networks

The use of neural networks always requires an adjustment to current problem conditions because it is hard to find a global adjustable model. However Souza's article [6] can be referred as it is an article in which a neural network model is applied for planning surgical correction of strabismus. In this case it was created a three layer network (*backpropagation* train) that predicts the need for surgical intervention.

Data Dimensions

Dimensionality of processing data must be considered when dealing a high amount of parameters. When that happens, the use of data dimension reduction methods (keeping the relevance of each element) can be required. This factor is not predictable during project initial stages; however, Edgar Ferreira and Paulo Barbeiro worked on a research project [7] (occurring at same time as project course) in which data reduction and clustering methods were studied.

4. Clinical Process Study

Before trying to improve the CCC clinic process, it was necessary to research on how it did happen at the time the project was started, through two different but essential points of view: patient and health professional.

At section 3.1 a description of existing computer systems was made and this chapter illustrates hierarchic and workflow dimensions of the clinic process, enhancing the role of persons, equipments and events.

4.1. Entities Relationship

The dimensions of clinical process analysis are shown below. Diagram 2 represents the hierarchical relation between persons and events on a generic patient consultation, emphasizing the clinical way, instead of the administrative way.

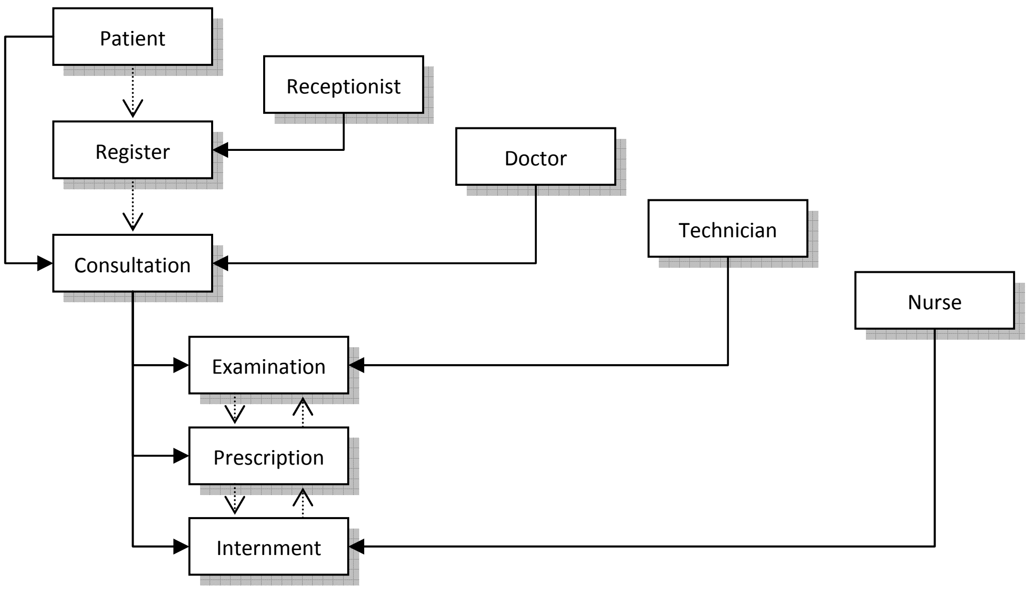


Diagram 2 - Hierarchical Relation in Clinical Process

The *Register* of a *Patient* before *Consultation* is executed only the first time the patient goes to CCC. The events *Examination*, *Prescription* and *Internment* are linked because one of them can cause the need for another.

Table 4 and Table 5 show, respectively, a brief description of each entity and event roles in clinical process.

Entity	Interactions
Patient	The patient is accepted for consultation or internment depending on disease, and doctor or service availabilities.
Doctor	Consultation doctors can have a consulting room inside or outside CCC. Surgeons are not CCC employees.
Technician	Examinations are usually made by a technician.
Nurse	The nurse is a CCC employee who takes care of interned patients.
Receptionist	The receptionist registers and guides the patient.

Table 4 - Entities Interactions in Clinical Process

Event	Interactions
Consultation	Consultation is scheduled in person or by phone.
Examination	The selection of an examination depends on disease. Examinations are made by request order.
Prescription	Drug prescription is made using a software tool that contains a drug database.
Internment	An internment can occur to perform a treatment or a surgery act.

Table 5 - Events Interactions in Clinical Process

4.2. Workflow

As it was already described, at CCC does not exist a common administrative and medical computer system, what makes clinical entities management a complex task. Furthermore, CCC building is separated on clinic and consultation sections that operate like independent companies with private doctors. These particularities generate large amounts of data and cause the replication of some tasks.

On this section it is intended to demonstrate the practical sequence of events happening in CCC. A special relief is given to the general workflow model and to the workflow model of doctors, who are the essential entities of the whole process.

4.2.1. Model

The main workflow model illustrates the functional relation among clinical process key events, including the intervenient entities and software. Curve arrows symbolize persons related to events where as dotted arrows identify types of utilized software.

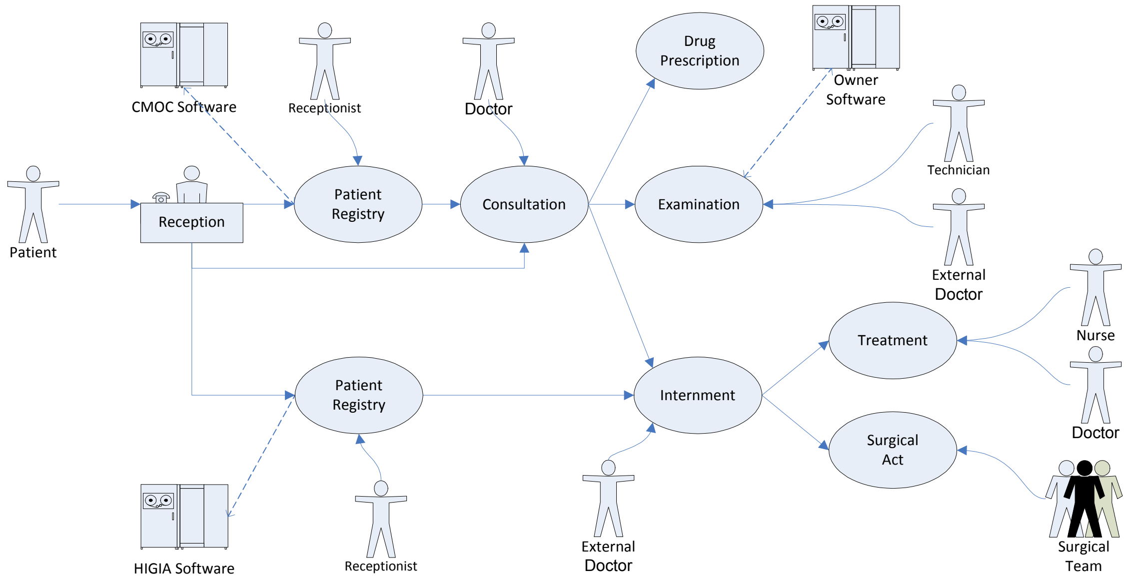


Diagram 3 - Main Workflow Model

4.2.2. Persons

On section 4.1 one can notice the interactions that occur among clinical process entities. On this section it is intended to reveal tasks attributed to each worker entity, putting them in CCC workflow.

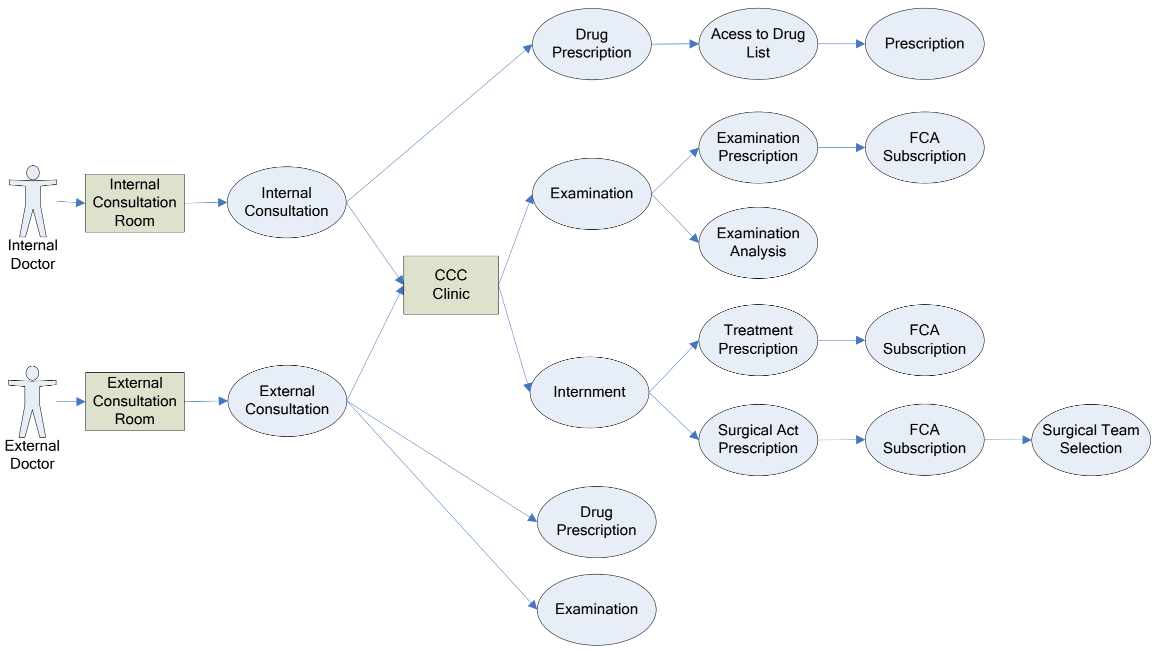


Diagram 4 - Doctor Workflow Model

The internal doctor makes consultations on a hired room inside CCC. The external doctor makes consultations outside CCC and these consultations can be complement with examinations and internments in CCC.

The formed surgical team depends on disease nature and doctor criterions. The patient's doctor can either be included on surgical team or not.

Before an examination, treatment or surgical act, patients or legal delegates must sign the Free and Clarified Assent (FCA) in order to authorize it.

Table 6 summarizes the main tasks performed by remaining workflow entities.

Entity	Tasks
Technician	- CMOC software management - Examinations - Compilation of treatment results for presentations
Nurse	- Consumer goods registry - Patient accessories registry - Clinical prescriptions registry - Hydro-electrolytic balance registry
Receptionist	- Patient registry in HIGIA - Patient registry in CMOC

Table 6 - Workflow tasks of several entities

5. Eye Diseases Study

For a project that intends to create Ophthalmologic diagnosis support tools, anatomic, physiologic and pathologic knowledge on this area is required. This is an engineering based task that does not require a medical like perception, however it needs a plain understanding of clinical decision subjects in order to develop data analysis solutions that generate medical expected results.

On this chapter, a compilation of the most important facts acquired on Ophthalmologic area is made, emphasizing the relationship between symptoms, convenient examinations, examination result features and diagnosis itself. Following sections seek to:

- Obtain the clearness cause/effect relationship for each clinical issue.
- Illustrate main characteristics of retinal diseases and required examinations.
- Specify retinal diseases that are OCT (Optical Coherence Tomography) detectable, as well as to specify measurable and relevant parameters on these images.

5.1. Overview

5.1.1. Anatomy and Diseases

The human eye can be divided into two main segments: the anterior segment and the posterior segment. The anterior segment is the front third of the eye that includes the structures in front of the vitreous humour: the cornea, iris, ciliary body, and lens. The posterior segment is the back two-thirds of the eye that includes the anterior hyaloid membrane and all structures behind it: the vitreous humour, retina, choroid, and optic nerve [8].

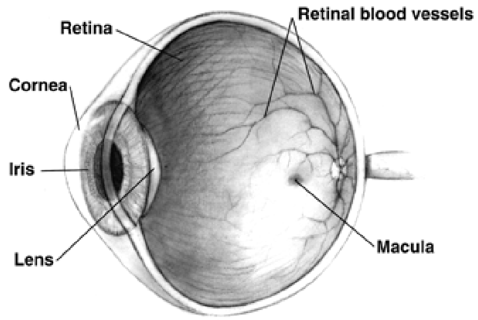


Figure 1 - Human Eye [8]

Table 7 illustrates the most significant diseases that affect both anterior and posterior eye segments. Obviously, it does not describe all known ocular diseases but a special attention is paid to the most common diseases and to the ones that bring on support interest.

Structure	Disease	Cause
Cornea	Cornea transparency decrease	Lack of eye lubrication caused by lipid secretion decrease
	Dry conjunctivitis	Inflammation or infection of the membrane lining the eyelids [9]
	Traumatism	Presence of a strange body
	Inflammatory reaction	
Lens	Cataract	Transparency decrease
Retina	Diabetic retinopathy	Bleeding of retinal capillaries Causes tearfulness
	Age-related macular degeneration (AMD)	Hardening of the arteries that nourish the retina [10]
	Retinal serous detachment	Separation of retina's sensory and pigment layers [11]
	Retinal pigment epithelium (RPE) detachment	Choroidal disorders that disrupt the normal junction between the basement membrane of the RPE and the inner collagenous layer of Bruch's membrane [12]
	Macular hole	Vitreous aging and hyaloids traction
	Epiretinal membrane	Cellular changes that occur spontaneously between the clear vitreous gel that is normally present, and the macula [13]
Optic Nerve	Glaucoma	Reduction of optic nerve neurons

Table 7 - Main Eye Diseases

5.1.2. Equipment Features

Table 8 demonstrates the main equipment characteristics and relates them to useful output parameters.

Equipment	Features	Output Parameters
Angiography	- Invasive examination; requires fluorescein or indocyanine injection. - Detects capillary leakage, bleeding, laser scars and abnormal vessels. [14]	- Image sequence showing marker dispersion through the vessels.
GDx™ Scanning Laser System	- Analyses neuron layer thickness around optic nerve.	- Colour coding image illustrating structural changes probability. - Linear probabilities plot for a retinal peripheral line.
Specular Microscopy	- Analyses cornea endothelium cells, usually to study surgery effects.	- Density and shift coefficient of cornea endothelium cells.
PEC (Perimetry)	- Evaluates the visual field projecting light dots on the retina. - Quantifies visual sensibility. [15]	- A visual field image illustrating decibel sensibility on each projection point.
Micro Perimetry	- Similar to PEC. - Uses laser projection to increase precision. - Evaluates blind points.	- A visual field image illustrating decibel sensibility on projection area.
VIS (Visumetry)	- Evaluates retina accuracy ignoring anterior segment defects. - Projects a laser letter on retina.	- A not dimensional number indicative of visual accuracy.
Stratus OCT (Zeiss)	- Evaluates macular area contour. - Detects several retina diseases like holes, detachments and macular edemas. [16]	- Radial line scans with identifying acquisition angle. - Macular thickness maps.
OCT-SLO (OTI)	- Similar to Stratus OCT. - Higher resolution and several scanning angles. [17]	- Longitudinal scan images. - Coronal CT scan images. - 3D topography images.

Equipment	Features	Output Parameters
Orbscan	- Measures cornea thickness.	- Colour coding images illustrating cornea thickness and statistics.
Retinography	- Analyses vessels stretching and bleeding.	- Colour or red component image of retina at high resolution.

Table 8 - Analysis of Examination Equipments Features

5.1.3. Study Issues

In order to remove a great part of medical procedure subjectivity, it is helpful to schematize the inherent events chain. Therefore, Diagram 5 illustrates three main study subjects, depending on patient initial condition (evaluation of symptoms and clinical history).

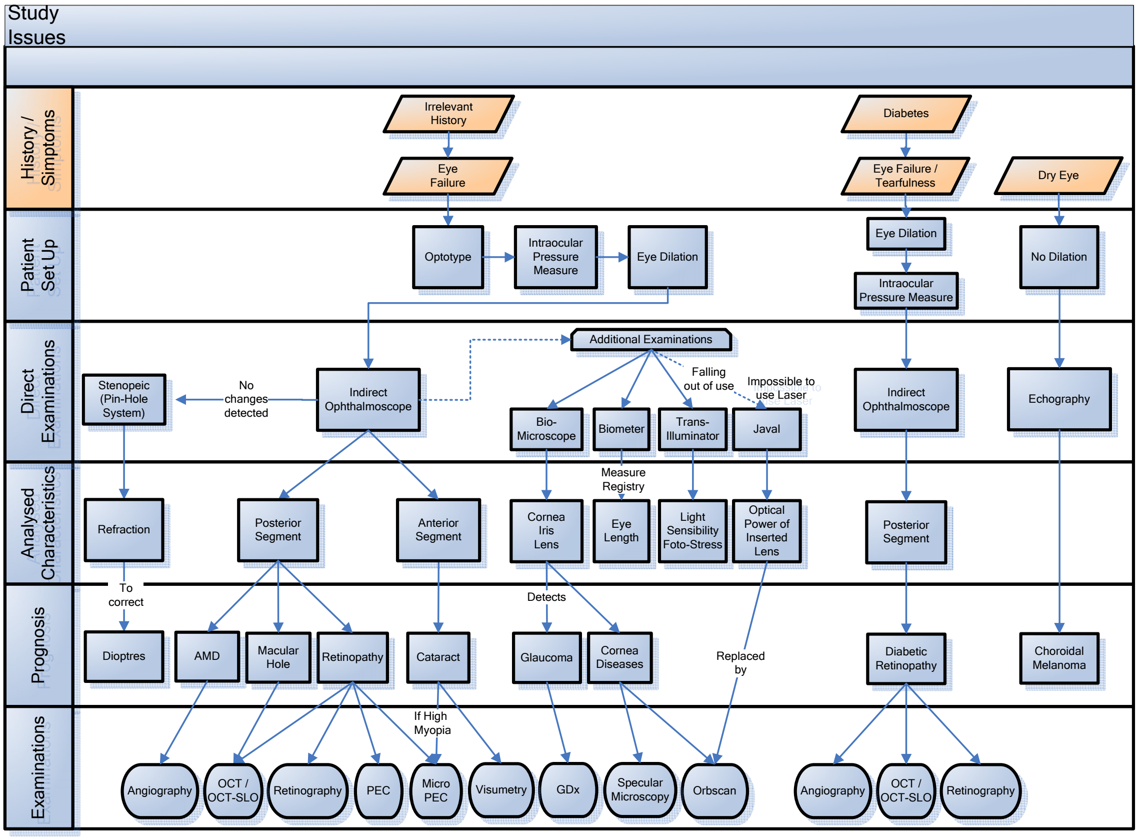
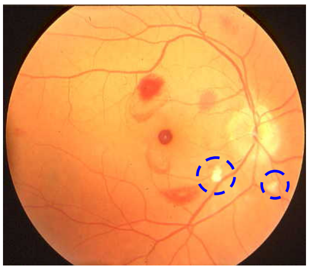
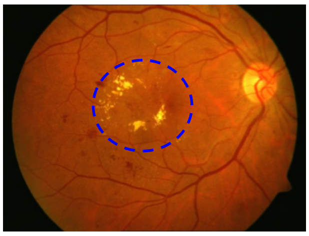
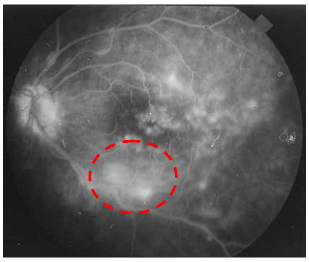
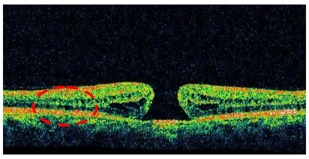
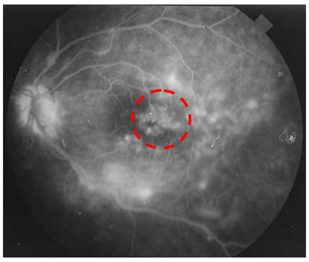
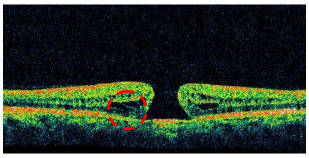


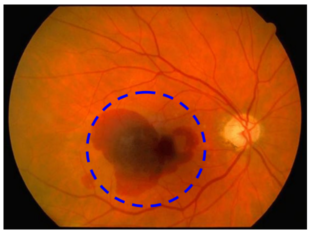
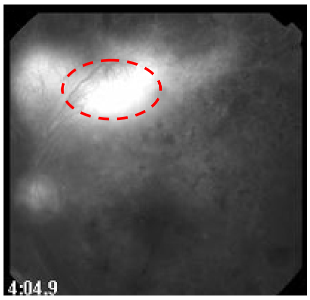
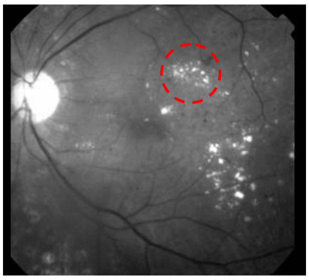
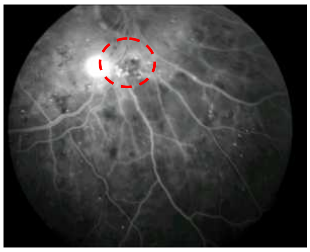
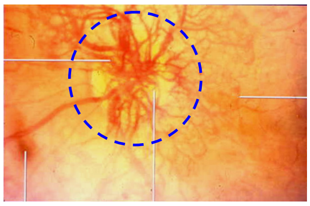
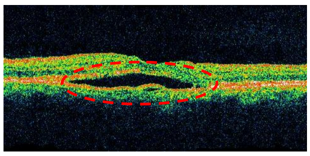
Diagram 5 - Clinical Chain of Study Issues

5.2. Retinopathies

This section introduces studied retinopathies, their related characteristics and diagnosis methodologies. A deeper study of retinal diseases was made because involved doctors consider these diseases as the most important ones for project purposes.

Table 9 shows study results. Characteristics of each retinopathy are described, as well as required examinations and observed deformities.

Disease	Examination	Characteristics	Image
Cotton-wool Exudates	Retinography	- White spots with smooth edges - Positioned above vessels	
	OCT	- Thickening of internal regions of retina - Hyper-reflective areas	
Hard Exudates	Retinography	- Yellow spots with distinct edges - Size similar to fovea - Radial shape around macula - Positioned above vessels	
Outer Cell Edemas	Angiography	- Fluorescent smooth area (grey colour) - Bigger then optic nerve	
	OCT	- Dark petal shaped spots - Positioned above RPE - Septum delimited	
Inner Cell Edemas	Angiography	- Similar to outer cell edemas - Positioned on macula area	
	OCT	- Similar to outer cell edemas - Positioned on macula area	

Disease	Examination	Characteristics	Image
Bleeding	Retinography	- Dark red regions - If superficial bleeding, exhibits round shape and optic nerve radius, at maximum	
	Angiography	- Hyper-fluorescent white regions	
Micro-Aneurisms	Angiography	- Little hyper-fluorescent white dots - Position coincident to vessels	
Retinal Neo-vessels	Angiography	- Diffuse white regions surrounding vessels	
Choroidal Neo-vessels	Retinography	- Abnormal reflective area due to RPE distortion - Occultation of sub-retinal membrane	
	Angiography	- Dispersed hyper-fluorescent regions when using green indocyanine	
Retinal Serous Detachment	Angiography	- Well delimited grey regions on final angiography stage	
	OCT	- Empty dark space above RPE	

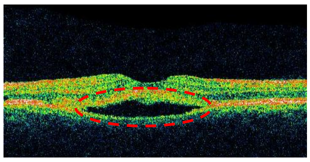
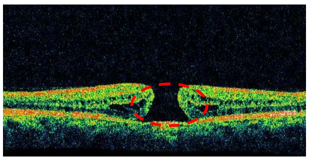
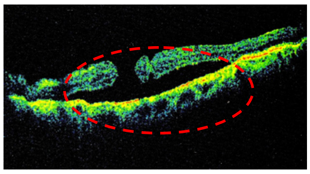
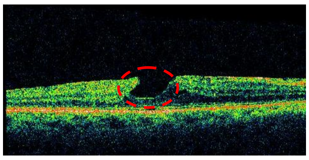
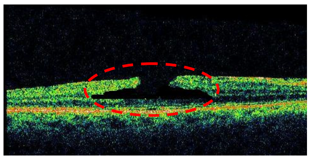
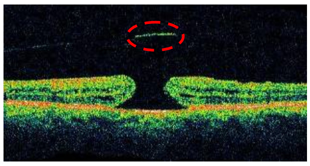
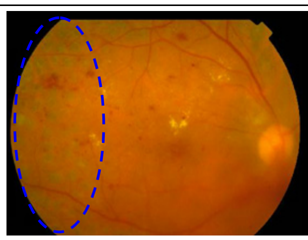
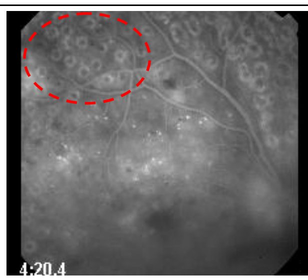
Disease	Examination	Characteristics	Image
Retinal Pigment Epithelium Detachment	Angiography	- Well delimited grey regions on final angiography stage	
	OCT	- Empty dark space below RPE	
Macular Hole	OCT	- Breach in superior retinal layer - Full retinal thickness is split	
Macular Hole with Detachment	OCT	- Breach in superior retinal layer - Separation between retina and RPE	
Lamellar Macular Hole	OCT	- Breach in superior retinal layer - Partial split of retinal thickness	
Lamellar Macular Hole with Detachment	OCT	- Breach in superior retinal layer - Retinal separation above RPE	
Epiretinal Membrane	OCT	- Part of superior retinal layer attached to hyaloid	
Laser Scars	Retinography	- Multiple grey-green rings	
	Angiography	- Multiple light grey rings with dark grey center	

Table 9 - Retinopathy Characteristics

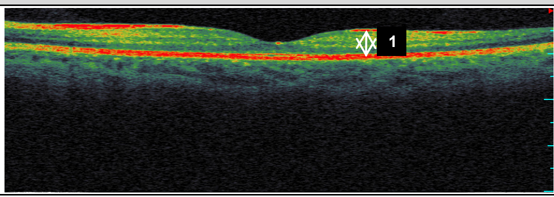
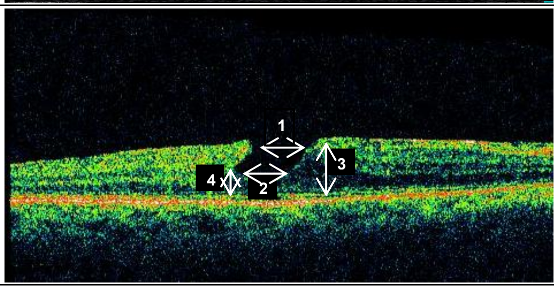
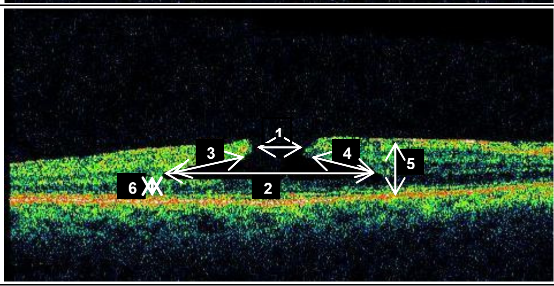
5.3. OCT Detectable Diseases

Measurable and discerning OCT image parameters that allow reaching a diagnosis are described on this section. The explained set of diseases was chosen because it is less complex and because, in most cases, only an OCT examination is required for achieving a correct diagnosis. This analysis includes retinal holes (macular and lamellar levels) and detachments (on several levels) diseases.

In the beginning of this project, the available OCT equipment was Stratus OCT, from Zeiss manufacturer; latter, the OCT-SLO equipment from OTI manufacturer was bought. These optical equipments use distinct acquisition geometries and generate images with different resolution. However, the type of produced images is similar and feature analysis is valid for both affairs.

Presently OCT-SLO is CCC preferred equipment for OCT images acquisition. This report contains images from distinct sources, but OCT and OCT-SLO images can be analyzed on a standard way, except when opposite indication is present.

Table 10 illustrates OCT detectable diseases, enhancing measurable and relevant parameters.

Disease	OCT Image	Relevant Parameters
Normal State		1. Retinal thickness 2. Acquisition angle
Lamellar Macular Hole		1. Hole aperture diameter 2. Hole base diameter 3. Retinal thickness 4. Hole-RPE thickness 5. Acquisition angle
Lamellar Macular Hole with Detachment		1. Hole aperture diameter 2. Detachment base diameter 3. Left arm length 4. Right arm length 5. Retinal thickness 6. Detachment-RPE thickness 7. Acquisition angle

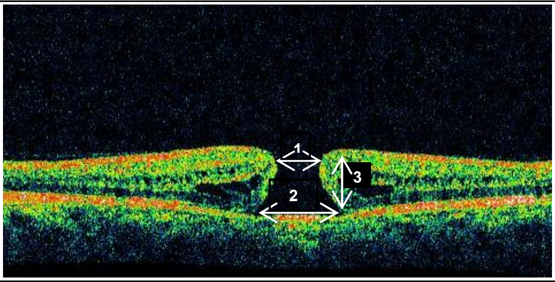
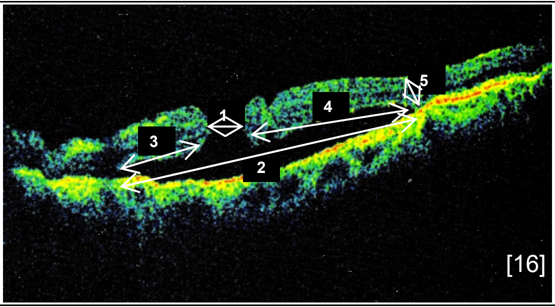
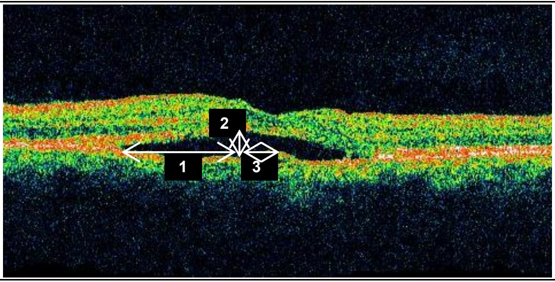
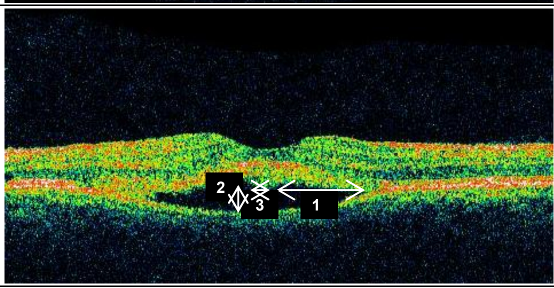
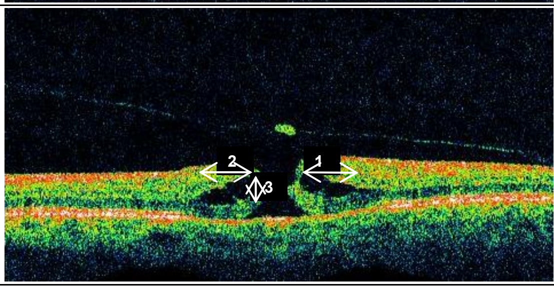
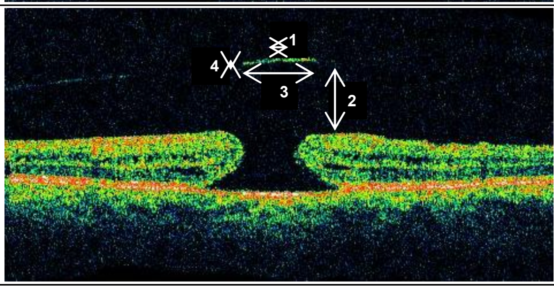
Disease	OCT Image	Relevant Parameters
Macular Hole		1. Minimum hole aperture diameter 2. Hole base diameter 3. Retinal thickness 4. Acquisition angle
Macular Hole with Detachment		1. Hole aperture diameter 2. Detachment base diameter 3. Left arm length 4. Right arm length 5. Retinal thickness 6. Acquisition angle
Retinal Serous Detachment		1. Detachment radius 2. Detachment central thickness 3. Distance to centre 4. Acquisition angle
Retinal Pigment Epithelium Detachment		1. Detachment radius 2. Detachment central thickness 3. Distance to centre 4. Acquisition angle
Edemas		1. Right side range 2. Left side range 3. Fraction of retinal thickness 4. Dispersion a. *Around hole edges* b. *Diffused* 5. Acquisition angle
Epiretinal Membrane		1. Distance to centre 2. Distance to retina 3. Width 4. Thickness 5. Acquisition angle

Table 10 - Significant Parameters in OCT Images

6. Software Development

Software development is an essential part of this project. In fact software component allows to integrate CCC heterogeneous computer systems as well as to provide data on an arranged and dynamic way. Otherwise, developing a proprietary solution will allow creating and integrating new modules that provide advanced functionalities.

However, despite of being an important part, software development is not the only component of this project. Thus, tasks scheduling was defined in order to allow an association of software development and other tasks.

Created application is not completed and needs an intensive set of tests before implementation on CCC or sale.

6.1. Architecture

Diagram 6 illustrates the architecture of required software application.

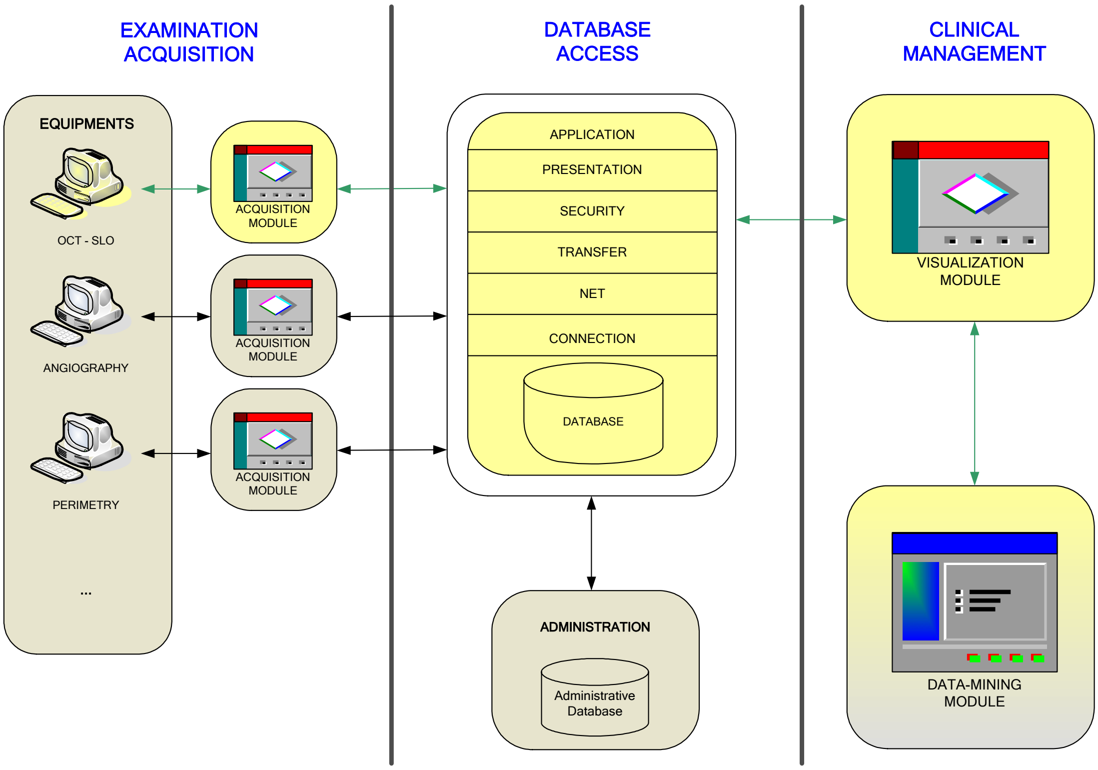


Diagram 6 - Software Architecture

As previous diagram illustrates, software architecture is divided in three main layers:

- Examination Acquisition: set of modules that allow importing and selecting images (and other parameters) from equipment software, as well as creating and saving examination reports.
- Database Access: this component contains the database where all doctor, patient and examination data are stored, allowing clinical management and intelligent processing (developed by Armanda Santos). This layer is accessed by a *virtual* web server (which will allow providing this service over the Internet). Health professional and patient info is imported from administrative software.
- Clinical Management: this component contains the module that allows consultation and examination management (developed by Paulo Barbeiro). Data-mining module is also part of this component.

The proposed architecture has several advantages on both functional and commercial points of view. Main advantages of this software architecture are:

- Acquisition, storing and processing of clinical data are supported by main DB
- Automatization of communication between examination equipments and main DB
- Ready for Internet access
- Synchronization between dedicated DB and administrative DB
- Fast access to clinical data
- Data sharing between modules
- Easy to integrate new modules or updates
- Module configuration allows partition of development stages
- Possibility of selling the product on separated elements

Components painted on yellow represent modules created in the course of this project. Data-mining module is on the first stage: image processing.

On this chapter both elements of software architecture developed by Edgar Ferreira are explained:

- OCT-SLO Acquisition Module (AM): module that imports images from OCT-SLO software and then allows user to select relevant images and to create an exam report.
- Main Interface (MI): main application that loads all software modules.

6.2. Requisites and Prototype

On this section features of AM and MI components are presented on separated diagrams (Diagram 7 and Diagram 8) that illustrate both user and functional requisites.

Since this is a medical solution, user interacting component must be enhanced. Thus, software must be user-friendly in order to captivate doctor interest.

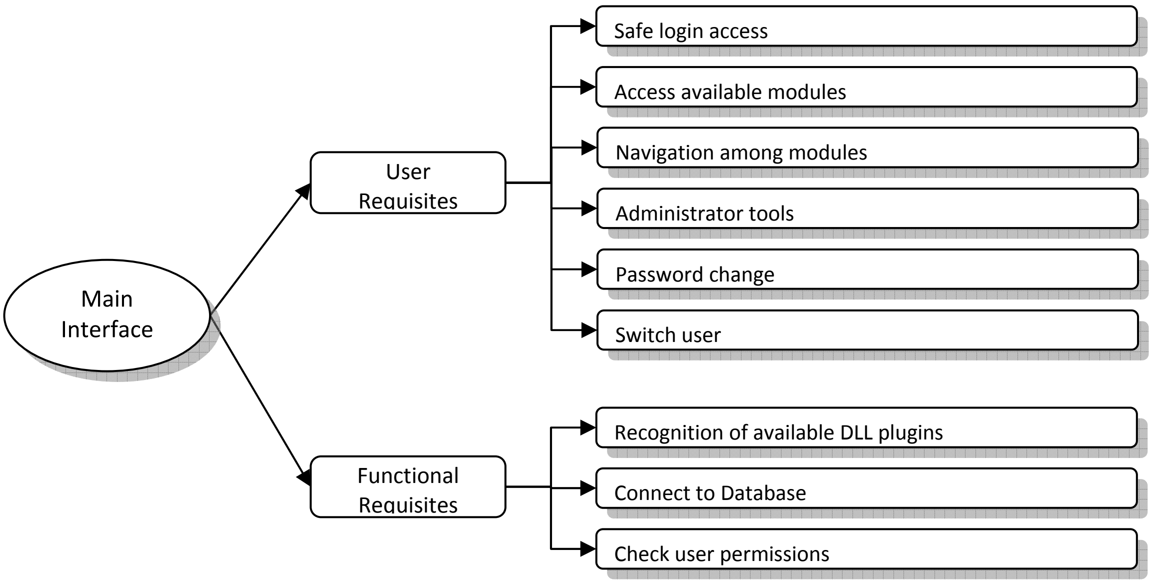


Diagram 7 - Main Interface Requisites

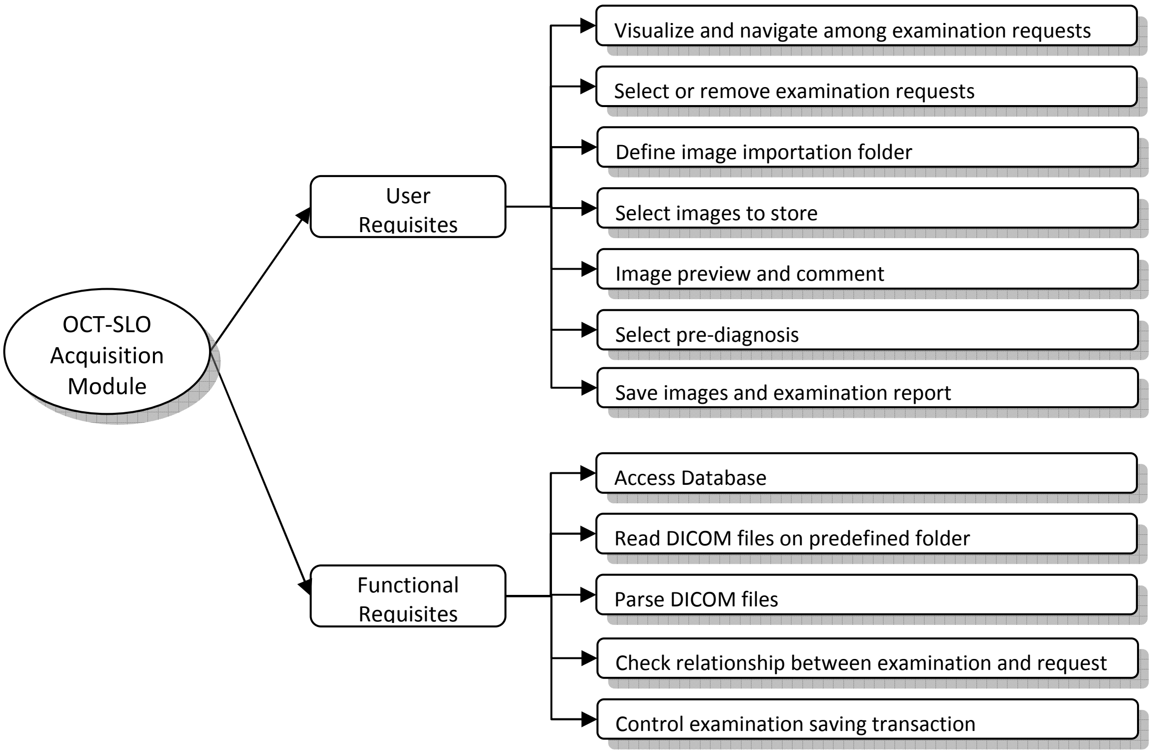
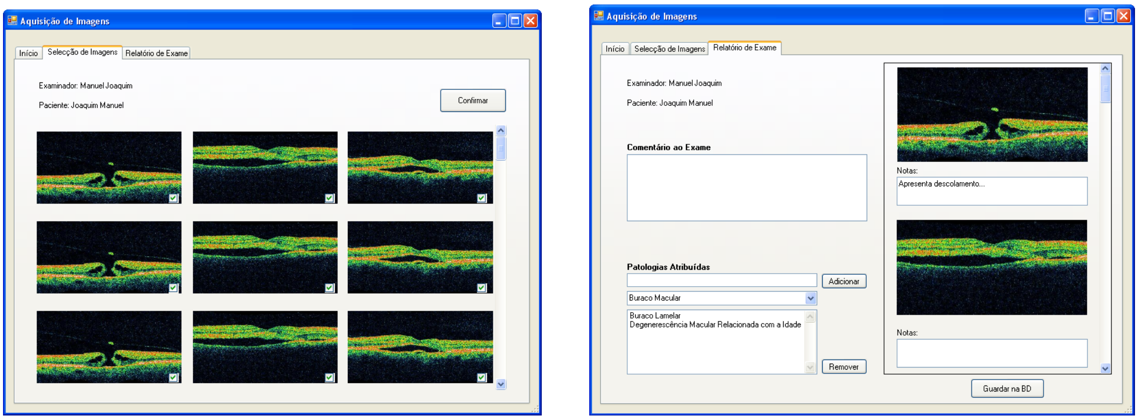


Diagram 8 - Acquisition Module Requisites

This solution was developed on C-Sharp (C#) language and Microsoft Visual Studio environment. C# is an object-oriented programming language developed by Microsoft which is based on C++ and includes aspects of several other programming languages (e.g. Java) with a particular emphasis on simplification [18]. Thus, Java programming experience acquired during Biomedical Engineering Course made easier adapting to this new programming language.

Another important requisite on software development is the creation of both Portuguese and English language versions to make possible a future wide expansion.

Figure 2 represents the created prototype of AM (the MI format was not defined at that time) that was approved by medical entities involved in the project.



a) b)

Figure 2 - Acquisition Module Prototype

As previous figure illustrates, there are two main user interfaces providing important features: a) selection of imported examination images; b) comment examination and selected images, select pre-diagnosis and save examination in Database (DB).

6.3. Implementation

6.3.1. Main Interface

Overview

MI is the Graphical User Interface (GUI) where all software features are available (including developed modules). Main class of this GUI is called *GlobalMainForm*.

On the next topics events and methods that allow MI features are described (as expressed requisites in Diagram 7). Figure 3 illustrates MI layout after a valid login; on left bar are located hyperlinks for available modules.

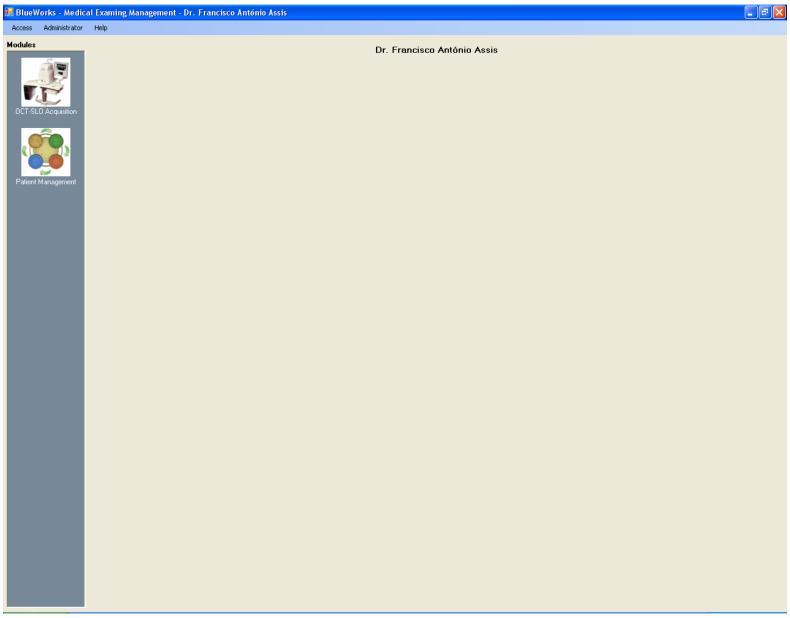


Figure 3 - Main Interface Overview

Plugin Interface

This solution uses plugin technology in order to easily create and load new modules providing new features. Available modules are defined by existing DLL files at each workstation.

However, developed plugins must implement several properties so they can be recognized by MI. Figure 4 illustrates the plugin interface (*IPlugin*) which is the responsible object for managing the connection between MI and AM (and other plugins).

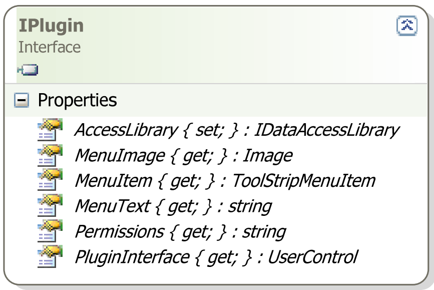


Figure 4 - Plugin Interface Properties

Table 11 describes each implemented property.

Property	Type	Description
AccessLibrary	*IDataAccessLibrary*	Interface between module and DB that allows calling of DB stored procedures and functions
MenuImage	*Image*	Icon sized image presented as module hyperlink
MenuItem	*ToolStripMenuItem*	Personalized item to be added to menu bar
MenuText	*string*	Plugin name that will appear as module hyperlink
Permissions	*string*	Allowed user types (separated by pipe - |)
PluginInterface	*UserControl*	Plugin GUI to be inserted on MI

Table 11 - Plugin Interface Properties

Initialization

Features involved in MI initialization are described on Table 12. After performing described tasks, MI acquires the layout represented on Figure 3.

Features	Description	Methods / **Objects** / *DB Functions*
Application initialization	When application starts graphical components are initialized and default actions are set.	`GlobalMainForm()` `InitializeComponent()` `GlobalMainForm_Load()` `ResetApplication()`
Database connection	A connection to DB is created using predefined connection string.	`ResetApplication()`
Login	Before MI is made available the login window is shown. User must enter a valid login or exit application.	`ResetApplication()` **`LoginForm()`** *`Login()`* *`GetDoctorName()`*
Load plugins	After valid login available plugin files (on Plugins folder) are read and valid modules are loaded. Module properties are implemented.	`ResetApplication()` `SavePluginsToList()`
Check user permissions	Each module permission string is parsed and only allowed modules are made visible to user.	`ResetApplication()` *`GetUserType()`*

Table 12 - MI Initialization Features

Figure 5 illustrates Login window.

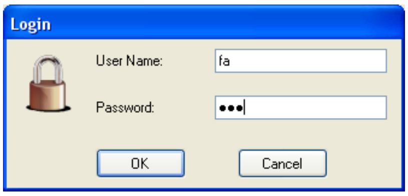


Figure 5 - MI Login Window

Load Modules

Features that perform module loading and navigation are described on Table 13.

Features	Description	Methods / **Objects** / *DB Functions*
Open module	When user clicks a module icon MI checks if that module is already opened. If not, respective *UserControl* is inserted on a new tab page.	`PanelEvent()`
Navigate among modules	If a module is already opened, clicking on the icon or tab title causes it to bring to front.	`PanelEvent()`
Close module	One can close current visible module by clicking close button. Plugins are reloaded in order to reset changes.	`buttonCloseTab_Click()` `SavePluginsToList()`
Open administrator module	Management module menu is available if user has administrator permission. This module is inserted on a new tab page but it is not loaded from a plugin.	`managementToolStripMenu` `Item_Click()` **`ManagementModule()`**

Table 13 - MI Load Module Features

Figure 6 illustrates MI layout when several modules are loaded. In this case, administrator management module is shown in front.

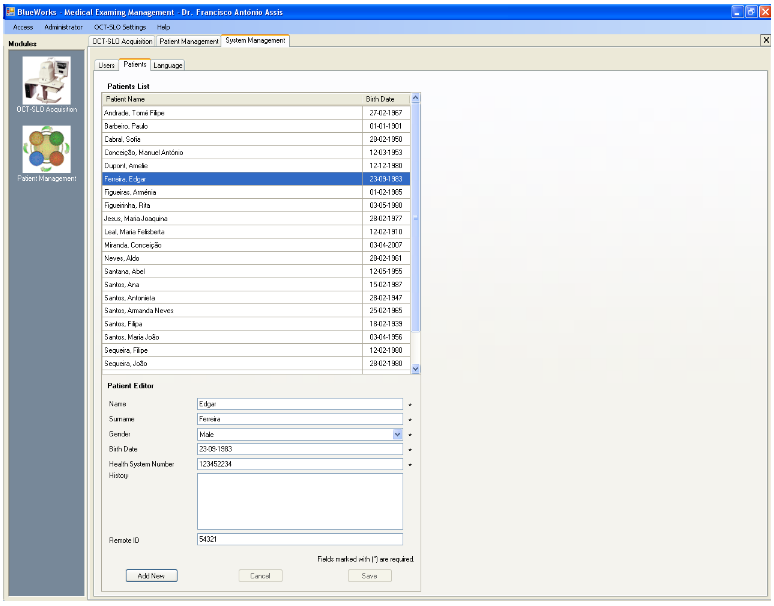


Figure 6 - Administrator Module Layout

Utilities

Table 14 describes available tools for application management. Management module tools are available only for administrator like users.

Features	**Description**	Methods / **Objects** / *DB Functions*
Change password	User can change his current login password by clicking the *Change Password* menu item. A new window is shown and user must retype a new password.	changePasswordToolStripMenuItem_Click() **ChangePasswordForm()** ChangePassword()
Logout	User can logout session. All resources are clean and login window is shown.	logoutToolStripMenuItem_Click() ResetApplication()
Exit application	User can exit application by four different ways: window close button, *Exit* menu item, Windows task bar and *Cancel* button on login window.	exitToolStripMenuItem_Click() GlobalMainForm_FormClosing()
Add or edit user	Administrator can edit or add a new user on *Users* tab page (management module).	buttonSaveUser_Click() *AlterUserData()* *AddNewUser()* UpdateUsersList() ShowUserPersonalData()
Add or edit patient	Administrator can edit or add a new patient on *Patients* tab page (management module).	buttonSavePatient_Click() *AlterPatientPersonalData()* *InsertPatient()* UpdatePatientList() ShowPatientPersonalData()
Change language	Administrator can change application language on *Language* tab page (management module). Available languages are Portuguese and English. Application must restart before changes are visible.	buttonChangeLanguage_Click()

Table 14 - MI Utilities

Figure 7 illustrates change password window.

Figure 7 - MI Change Password Window

6.3.2. OCT-SLO Acquisition Module

AM is the module that performs the connection between this solution and OCT-SLO proprietary software. This module main tactic is to import DICOM (Digital Imaging and Communications in Medicine) files exported by equipment software, extracting respective images and then allow user to choose and save the more relevant images.

Overview

AM main class is called *MainContents*. On the next topics events and methods that allow AM features (as expressed requisites in Diagram 8) are described. Figure 8 illustrates AM initial layout (showing all OCT-SLO examination requests).

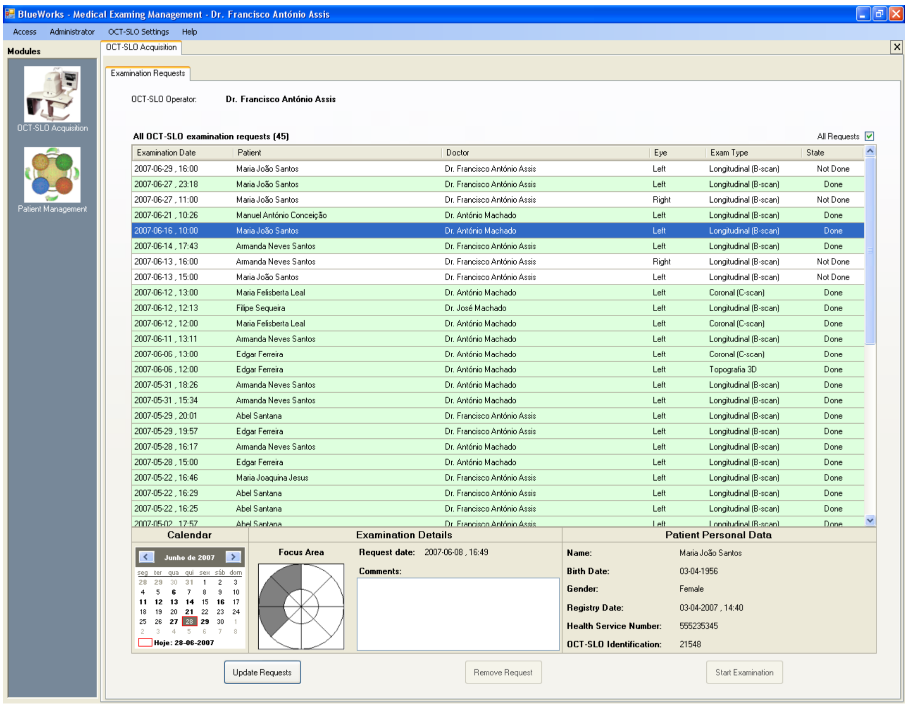


Figure 8 - AM Initial Layout

Initialization

Features involved in AM initialization are described on Table 15. After performing described tasks, MI acquires a layout similar to illustration of Figure 8 (only not performed examination requests are shown).

Features	**Description**	Methods / **Objects** / *DB Functions*
Module initialization	AM initialization occurs on two distinct phases: - Plugin file read (*MainContents* constructor is called and respective *UserControl* is initialized); - Module load (when AM is made visible in MI tab control, DB connection object has already been sent to *MainContents*; AM menu item is inserted at this time).	`MainContents()` `InitializeComponent()` `MainContents_Load()`
Set calendar bolded dates	Calendar object is updated in order to bold all examination request dates (only not done examinations).	`UpdateCalendarBoldedDates()` *`GetNotDoneExams()`*
Show examination requests on table	Not done examinations at selected Month Calendar date (current day by default) are shown on requests table. If there are no examination requests for selected date, all not done examinations are shown.	`UpdateRequestsToSelectedDate()` `UpdateRequestTable()` *`GetNotDoneExams()`*
Show current patient and examination request data	Last examination request (first table row) is selected by default. Patient and examination details are shown on graphical objects below requests table.	`dataGridRequests_SelectionChanged()` *`GetPatientPersonalData()`* *`GetEquipmentMatchID()`*

Table 15 - AM Initialization Features

Examination Requests Management

Table 16 describes features involved on requests management. These functionalities are available on *Examination Requests* tab page.

Features	**Description**	Methods / **Objects** / *DB Functions*
Change type of examination requests visualization	User can alter requests table visualization type by changing *All Requests* checkbox state. If this checkbox is selected all examination requests are shown (even performed and deleted ones). Performed examinations are painted on green and deleted requests are painted on red.	`checkBoxAllRequests_CheckedChanged()` `UpdateCalendarBoldedDates()` *`GetNotDoneExams()`* *`GetAllExamRequests()`* `UpdateRequestsToSelectedDate()` `UpdateRequestTable()`
Select calendar date	Not done examinations at selected Month Calendar date are shown on requests table. If there are no examination requests for selected date all not performed examinations are shown. If *All Requests* is checked, nothing changes.	`monthCalendar_DateChanged()` `UpdateRequestsToSelectedDate()` `UpdateRequestTable()`
Update examination requests table	User can update examination requests (keeping current visualization settings) by clicking *Update Requests* button.	`buttonUpdate_Click()` `UpdateCalendarBoldedDates()` *`GetNotDoneExams()`* *`GetAllExamRequests()`* `UpdateRequestsToSelectedDate()` `UpdateRequestTable()`
Show selected patient and examination request data	Examination requests can be selected by clicking table rows. Selected patient and examination details are shown on graphical objects below requests table.	`dataGridRequests_SelectionChanged()` *`GetPatientPersonalData()`* *`GetEquipmentMatchID()`*

Features	Description	Methods / **Objects** / *DB Functions*
Remove examination request	User can remove an examination request by selecting it and clicking *Remove Request* button. Selected request is not physically removed from DB but is invalidated.	buttonRemoveExam_Click() *InvalidateExamRequest()* UpdateCalendarBoldedDates() *GetNotDoneExams()* *GetAllExamRequests()* UpdateRequestsToSelectedDate() UpdateRequestTable()
Select examination request	User can select an examination to perform by double-clicking a table row or clicking *Start Examination* button. Performed or deleted examinations are not available. A notifying dialog window is shown until images importation or user abort.	dataGridRequests_CellMouseDoubleClick() buttonMakeExam_Click() OpenWaitingDialog() **WaitingDialog()**

Table 16 - AM Examination Requests Features

Figure 9 illustrates waiting dialog window. This message indicates that application is waiting for image exportation on watched folder.

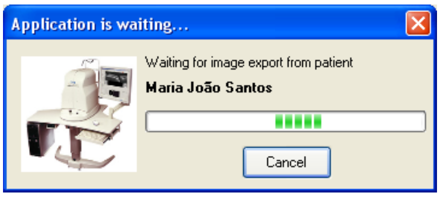


Figure 9 - AM Waiting Dialog

Images Selection

Table 17 describes features involved on image import and selection.

Features	Description	Methods / **Objects** / *DB Functions*
Define image importation folder	The watched folder (where OCT-SLO should put examination images) can be defined by user (using *Image Folder* item on *OCT-SLO Settings* menu) on a folder browser.	folderMenu_Click()
Read images	When user exports OCT-SLO images to watched folder, AM detects the existence of a new image folder (DICOM format) and implements file reading cycles. DICOM files are parsed using a class based on *dicomcs* [19] parser and image thumbnails are shown on *Imported Images* area (*Image Selection* tab page). Files are imported even if user did not choose an examination request.	exportFolderWatcher_Changed() **DicomReader()** **SelectionThumbnail()**
Check examination consistency	When the first image is imported, several consistency checks are performed: - Patient OCT-SLO identification number (ID) is read from DICOM file and compared with matching database ID (if it already exists); - If matching database ID does not exist, byrthdates from DB and DICOM file are compared; - Laterality field from DICOM file is compared with request eye.	exportFolderWatcher_Changed() *GetEquipmentMatchPatientID()*

Features	Description	Methods / **Objects /** *DB Functions*
Validate patient	If previous consistency check did not get a patient match, *Patient* combobox is shown and user must select the patient who has performed current examination. If selected patient has any not done examination request, user can select it from shown window.	`exportFolderWatcher_Changed()` `comboBoxAllPatients_SelectedIndexChanged()` `OpenPatientMatchWindow()` *`GetNotDoneExamsByPatientByEquipment()`* **`PatientMatchWindow()`**
Select images	User can select or unselect an image by clicking on *Select Image* check box (inside respective thumbnail). It is also possible to select or unselect all imported images at a time, by clicking *All Images* check box.	**SelectionThumbnail:** `checkImage_CheckedChanged()` `ChangeState()` **MainContents:** `checkBoxAllImages_CheckedChanged()` `UncheckAllImagesBox()`
Use images	After selecting images, user can send them to next tab page (*Examination Report*). Any existing images on last tab page are replaced by the new set of selected pages (editable thumbnails). If no examination request has been selected, user must define examination type and demander. Pre-diagnosis combo box is filled with all DB diseases.	`buttonConfirm_Click()` **`EditionThumbnail()`** *`GetAllHealthProfs()`* *`GetAllExamTypes()`* *`GetAllPatologies()`* `comboBoxAllDoctors_SelectedIndexChanged()`

Table 17 - AM Image Selection Features

Figure 10 illustrates *Image Selection* tab page. In this case the correspondence between selected request and performed examination was not detected. Thus, *Patient* combo box is enabled and user must select a patient before choosing relevant images. Image thumbnails shown at *Imported Images* area are *SelectionThumbnail* objects.

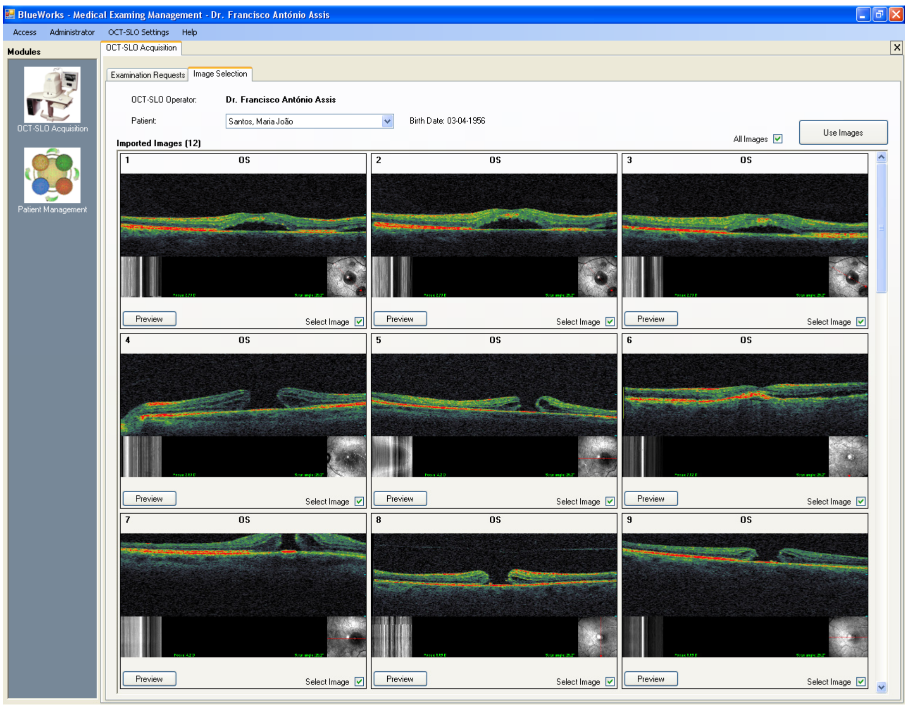


Figure 10 - AM Image Selection Tab Page

Figure 11 illustrates *Examination Requests* window which is shown when user selects a patient who has undone examination requests.

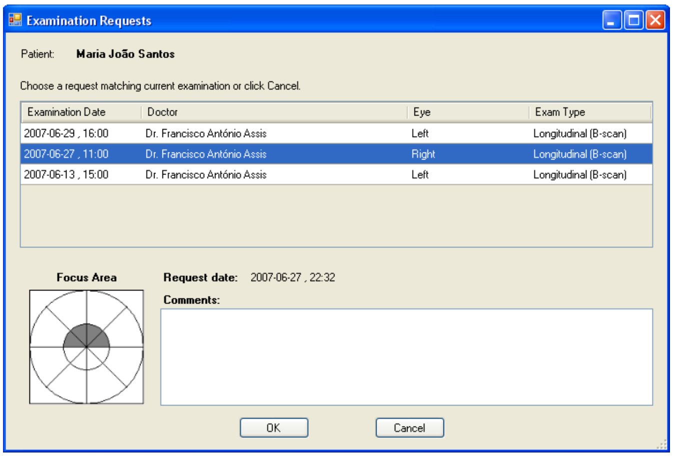


Figure 11 - AM Examination Requests Window

Examination Report

Table 18 describes examination report defining and saving features.

Features	Description	Methods / **Objects** / *DB Functions*
Preview and comment images	User can add different comments to each selected image by typing on *Insert notes* field. An image preview window is shown when user clicks *Preview* button or double-clicks on image. On this window it is also possible to insert image comments. Examination comments can be typed on *Examination Comments* field.	**EditionThumbnail:** buttonPreview_Click() imageBox_DoubleClick() **ImagePreview()**
Save image as a file	User can save the preview image on several formats (PNG, JPEG, BMP and GIF) by clicking *Save Image* item on *File* menu.	saveImageToolStripMenuItem1_Click()
Select pre-diagnosis	Several pre-diagnosis can be selected at *Pre-Diagnosis* combo box and added by clicking *Add* button. To limit combo box list, user can type part of the disease name at *Pre-Diagnosis* edit area. Added diseases can be either removed from list box by clicking at *Remove* button.	textTiped_TextChanged() buttonAdd_Click() buttonRemovePreDiagnosis_Click()
Save images and examination report	Examination images and report are saved on DB by clicking *Save Examination* button. Several checks are performed at this time: - A progress bar demonstrates saving progression; - Inserted comments are formatted in order to include user identification; - A database transaction is started; - After saving examination and images the transaction is *commited*. If any error occurs in the course of this process the transaction is *rollbacked*; - Calendar and requests table are updated (*Examination Requests* tab page).	buttonSaveExam_Click() *StartTransaction()* *CreateExamOCT()* *InternInsertImage()* *CommitTransaction()* *RollbackTransaction()* UpdateCalendarBoldedDates() UpdateRequestsToSelectedDate()

Table 18 - AM Examination Report Features

Figure 12 illustrates *Examination Report* tab page. In this case the correspondence between selected request and performed examination was not detected. Thus, *Examination demander* and *Examination Type* combo boxes are enabled and user must confirm these fields before saving examination. Image thumbnails shown at *Selected Images* area are *EditionThumbnail* objects.

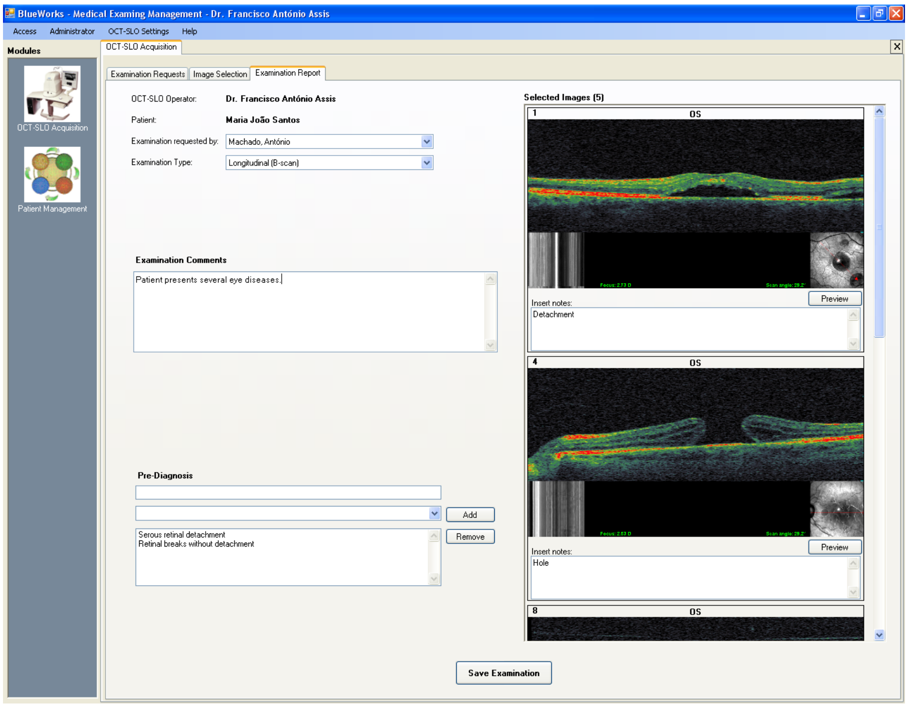


Figure 12 - AM Examination Report Tab Page

Figure 13 illustrates *Image Preview* window which is shown when user double-clicks an image (or clicks *Preview* button) at *Selection* or *Edition Thumbnails*. Image zoom mode can be changed on this window and comments can be inserted.

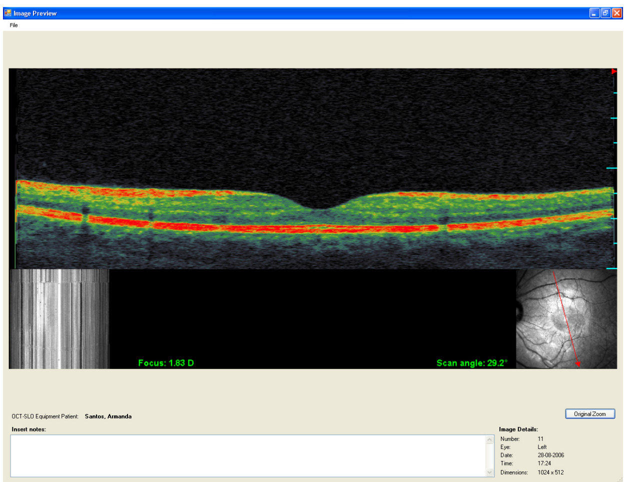


Figure 13 - AM Image Preview Window

6.4. Challenges

Several challenges and obstacles appeared in the course of software development. Thus, a great part of time resources were given to these tasks. These difficulties can be explained by application specificity (there are no conceptual models to follow) as well as the lack of Software Engineering experience (Informatics Engineers working at ISA gave a crucial support on this area). Table 19 describes main surpassed challenges in the course of software development.

Features	Challenges	Solutions
C# and Visual Studio familiarization	Despite previous experience using Python, Matlab and Java programming languages, the adaptation to a new language needs a learning period. Making GUIs using object-oriented programming languages was also a new challenge.	The orientation of ISA's Informatics Engineers was very important in order to surpass this obstacle.
Plugin management	The use of DLL provided modules is a task that implicates programming behaviour changes because separated applications are created and integrated.	A plugin interface demo from ISA software archives was provided. Several required adaptations were made.
File importation	Reading files in real time implicates safe methods. Several exceptions occur when trying to load a partial file (if the file is not completely created).	A try-catch cycle was created in order to prevent file reading errors. This cycle stops when the file is complete and ready for loading.
DICOM parser	The integration of an external application always requires some adaptations. Furthermore DICOM parsing is not a trivial task and requires many interpretation methods.	Created class *DicomReader* uses main *dicomcs* parsing system (this solution was integrated on AM) in order to get the file *dataset*.
DICOM image recognition	The DICOM image field is not a C# recognizable format. Thus, getting imported image is not an direct task.	The stages of implemented DICOM image recognition system are: - Get image parameters from DICOM dataset; - Create and menage a bitmap image directly in system memory (using pointers); - Pixel values are defined by reading DICOM pixel data field (byte by byte). [20]
Retinal focus tool	This tool (available at Patient Management Module) implicates bitmap manipulation and delimitation of image regions.	Like on DICOM image recognition it was necessary to create unsafe code (pointers to system memory). When user clicks a region, all contained pixels are painted in grey.
DB transaction control	The examination saving method must be able to guarantee that all data are saved. If an error occurs, the process must be completely anulled.	As explained on Table 18 a DB transaction is created in order to control saving flow.
Creation of the English version	When creating a different language version, all text fields and dialog boxes must be changed. It is also required a language management tool.	Stages of English version creation were: - Translate all GUI text objects; - Create a resources object that includes both Portuguese and English string versions; - Create the *Change Language* tool on *System Management* module; Resources DLL files must be also available when MI loads plugins.

Table 19 - Software Development Challenges

6.5. Software Tests

One verified software component limitation is the difficulty of scheduling and executing tests to created modules. On this particular case, the lack of Software Engineering experience may have negatively influenced developed work.

However, a set of tests was performed on MI and AM modules in order to check the ability to execute required tasks as well as to check application behaviour under anomalous conditions. Tests were performed under user points of view (net connection state and several conditions were controlled). Not all required tests were performed because of time limitations.

Table 20 describes tests conditions and results.

Module	Feature	Functionality	Conditions	Net	Result	Approval
MI	Login	Try to exit application and then return to login		✓	Returns to login	OK
MI	Login	Try to exit application and then return to login		✗	Returns to login	OK
MI	Login	Try to confirm with empty fields		✓	Exits application	OK
MI	Login	Exit application		✗	Exits application	OK
MI	Login	Try to login with empty fields		✓	OK button is disabled	OK
MI	Login	Try to login with invalid username		✓	Warning Message	OK
MI	Login	Try to login with invalid username		✗	Warning Message	OK
MI	Login	Try to login with invalid password		✓	Warning Message	OK
MI	Login	Try to login with invalid password		✗	Warning Message	OK
MI	Login	Try to login with both invalid fields		✓	Warning Message	OK
MI	Login	Try to login with both invalid fields		✗	Warning Message	OK
MI	Login	Valid login		✓	Enter MI	OK
MI	Login	Valid login		✗	Warning Message	OK
MI	Logout	Logout		✓	Returns to login	OK
MI	Logout	Logout		✗	Returns to login	OK
MI	Change Password	Try to confirm with empty fields		✓	Warning Message	OK
MI	Change Password	Try to confirm with invalid password		✓	Warning Message	OK
MI	Change Password	Try to confirm with invalid retyped password		✓	Warning Message	OK
MI	Change Password	Try to confirm with valid fields		✓	Changes password	OK
MI	Change Password	Try to confirm with valid fields		✗	Warning Message	OK
MI	Exit	Try to confirm with empty fields		✓	Exits application	OK
AM	Module Loading	Paste an image folder on watched folder	*Module not loaded*	✓	Nothing	OK
AM	Module Loading	Left click on module icon	*Module not loaded*	✓	Opens AM	OK
AM	Module Loading	Right click on module icon	*Module not loaded*	✓	Nothing	OK
AM	Requests Navigation	Show requests on table	*No requests on DB*	✓	Empty table	OK
AM	Requests Navigation	Switch calendar days	*No requests on DB*	✓	Empty table	OK
AM	Requests Navigation	Switch All Requests mode	*No requests on DB*	✓	Empty table	OK
AM	Requests Navigation	Update requests	*No requests on DB*	✓	Empty table	OK
AM	Requests Navigation	Update requests		✓	Warning Message	OK

Module	Feature	Functionality	Conditions	Net	Result	Approval
AM	Requests Navigation	Show requests on table		✓	Requests are shown	**OK**
AM	Requests Navigation	Switch calendar days		✓	Requests for date are shown	**OK**
AM	Requests Navigation	Switch All Requests mode		✓	All requests are shown	**OK**
AM	Requests Navigation	Update requests		✓	Updates requests	**OK**
AM	Requests Navigation	Select a different request		✓	Shows request and patient details	**OK**
AM	Requests Navigation	Remove request		✓	Removes request. Appears as deleted on All Requests mode	**OK**
AM	Requests Navigation	Start Examination		✓	Opens Waiting Dialog	**OK**
AM	Requests Navigation	Double-click on not done examination request		✓	Opens Waiting Dialog	**OK**
AM	Requests Navigation	Double-click on removed examination request		✓	Warning Message	**OK**
AM	Requests Navigation	Double-click on done examination request		✓	Warning Message	**OK**
AM	Requests Navigation	Remove request	*No line selected*	✓	Nothing	**OK**
AM	Requests Navigation	Start Examination	*No line selected*	✓	Nothing	**OK**
AM	Requests Navigation	Reorder columns	*Not done requests*	✓	Columns are ordered	**OK**
AM	Requests Navigation	Reorder columns	*All requests*	✓	Columns are ordered. Colours are kept	**OK**
AM	Import Images	Paste an image folder on watched folder	*Matching request*	✓	Imports images ant sets request parameters	**OK**
AM	Import Images	Paste an image folder on watched folder	*Matching request but wrong eye*	✓	Imports images ant sets request parameters. Warning message	**OK**
AM	Import Images	Paste an image folder on watched folder	*No selected request*	✓	Imports images. Asks for patient selection. Selects patient by DB compare	**OK**
AM	Image Selection	Double-click on selection thumbnail		✓	Opens Preview Window with disabled notes	**OK**
AM	Image Selection	Click on preview button		✓	Opens Preview Window with disabled notes	**OK**
AM	Image Selection	Click on All Requests check box		✓	Selects / unselects all thumbnails	**OK**
AM	Image Selection	Click on Select Image check box		✓	Selects / unselects selected thumbnail	**OK**
AM	Use Images	Click on Use Images button	*Matching request*	✓	Opens Examination Report tab page. Sets request parameters	**OK**
AM	Use Images	Click on Use Images button	*No matching request*	✓	Opens Examination Report tab page. Asks for request parameters selection	**OK**
AM	Save Examination	Click on Save Examination button		✓	Shows progress bar. Examination is saved.	**OK**

Table 20 - Software Tests

7. Image Processing

Image processing is an important part of this project. Although business planning suggests the improvement of software modules before creating a data-mining module, the academic role of this project prevails and a significant fraction of available course time was invested on image processing for features detection.

As section 5.3 suggests, OCT images allow detecting several retinal diseases (including retinal holes and detachments) with high success levels. Created algorithms automatically detect important parameters for retinal diseases categorization.

This chapter separates image processing in three main parts:

- Definition of algorithm purposes
- Algorithm implementation stages
- Algorithm validation

7.1. Requisites

A large fraction of OCT detectable diseases occurs in the region surrounded by Retinal Anterior Limit (RAL) and superior limit of Retinal Pigment Epithelium (RPE). RAL is the virtual membrane that separates retina from vitreous space and RPE limit matches RPE internal surface that is represented by a red line.

An ophthalmologist is able to immediately detect these elements because he has the capability to perceive the image as a whole. However, at computer analysis level, this ability does not exist. In fact the big purpose at this time is to overcome this challenge.

Image processing algorithms were developed in Matlab. This is a high-level language that includes a useful image processing toolbox and allows the integration of a free runtime instance on developed C# application.

Matlab is widely employed during Biomedical Engineering Course, so that it does not require an extensive search on programming language details.

Diagram 9 illustrates main identified requisites for developed image processing algorithms.

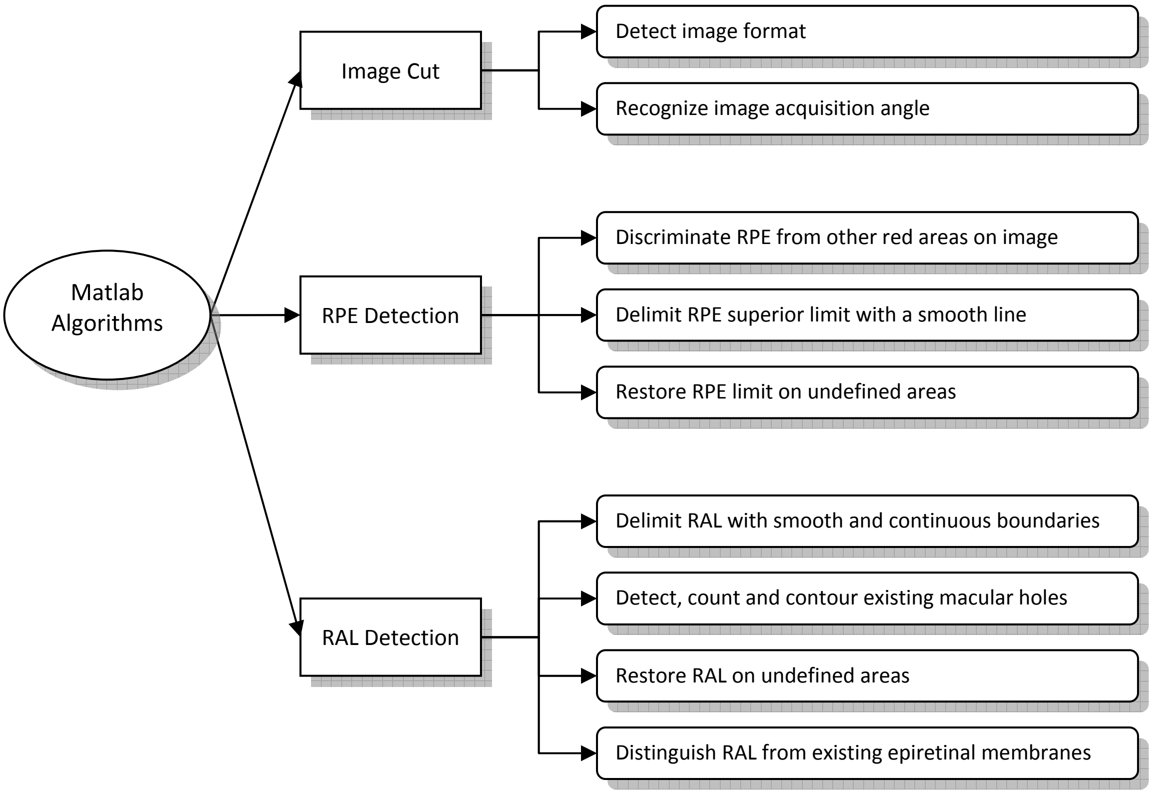


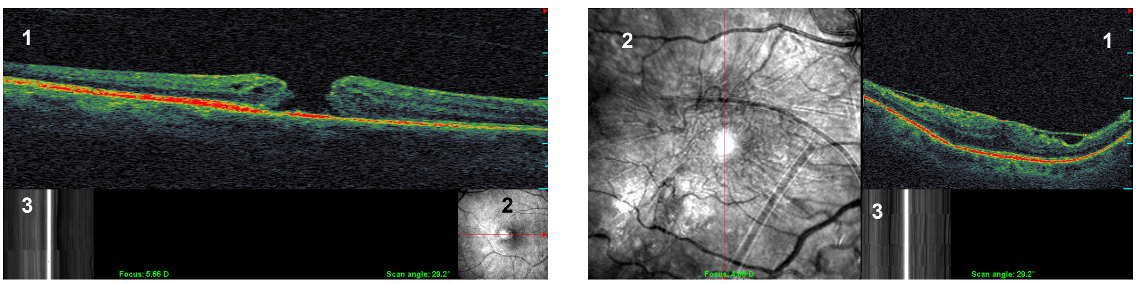
Diagram 9 - Image Processing Requisites

7.2. Implementation

7.2.1. Image Cut

OCT images are acquired from both Stratus OCT and OCT-SLO equipments on B-Scan acquisition method.

As Figure 14 illustrates, OCT-SLO images can have two different element configurations.



a) b)

Figure 14 - OCT-SLO Acquired Images

Images Figure 14.a) and Figure 14.b) illustrate the same type of examination with distinct arrange of elements: retinal line scan (1); retinography with acquisition direction arrow (2); alignment of deep scans (3).

OCT images are exported one by one and do not contain these additional elements.

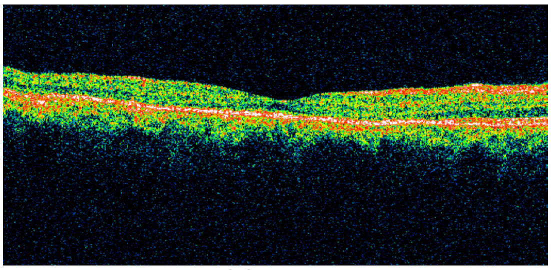


Figure 15 - OCT Acquired Image

Format Detection

Matlab function *CutImage* was developed in order to detect OCT-SLO image main elements. This function gets the original image matrix, detects its format and returns line scan section and acquisition angle. OCT images do not need any previous treatment.

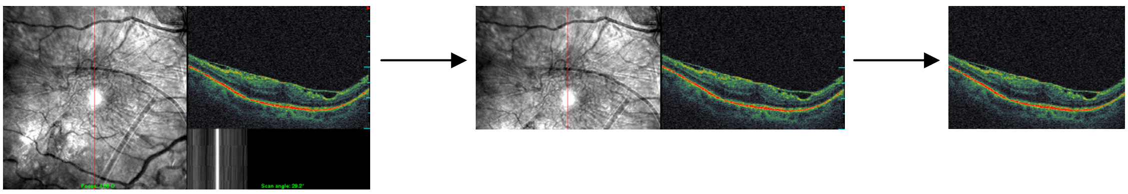
```
[imageOut angleOut] = CutImage(imageIn);
```

Distinction between OCT-SLO (Figure 14) and OCT (Figure 15) images is based on their size. In other words, if image dimension is not 1024 columns by 512 rows, then it is an OCT image (function returns the same image and angle equals to zero). If dimension is (1024, 512), the image is processed as described below.

Both pictures in Figure 14 have in common their useful height: 342 rows from top. All OCT-SLO recognized images are cut at the 342nd row.

To prevent rule lines from interfering on image processing, the last 18 columns are removed. In situations as illustrated on Figure 14.b) it is necessary to remove the left half of image (Figure 16):

- Pixels are analysed from left to right in order to analyse colour depth;
- If column 513 is the first RGB column (RGB components not equal), image is removed below this index.



a) b) c)

Figure 16 - Image Cut Stages. a) Original Image. b) Horizontal Cut. c) Vertical Cut.

Angle Recognition

For acquisition angle recognition it is necessary the analysis of segment (3), shown at Figure 14. The angle detection algorithm is based on direction arrow coordinates and performs next stages:

- Arrow detection: only coloured pixels are kept (greyscale points are removed);
- Arrow initial extremity (x_1, y_1): first white pixel found on left to right scan;
- Arrow final extremity (x_2, y_2): last white pixel found on left to right scan;
- Arrow pointer: when 3 to 5 white pixels are found on a single column;
- Arrow orientation: when not a vertical arrow, the formula $\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$ is applied;
- Arrow direction: the angle is fixed at 0º to 360º range by checking arrow pointer position.

7.2.2. Retinal Pigment Epithelium Detection

RPE detection algorithm is based on three main methods: spatial filtering, snake filtering and scanning selection. These methods are performed on function *GetEpithelium*. This function gets original image and several processing parameters (if parameters fields are empty, best found values are applied) returning RPE coordinates vector and images from both filtering and scanning stages.

```
[epithelium img_filtered img_display] = GetEpithelium(display, display_final, img,
                                        filename, kernel, threshold, neig, meanR, blur, borderRange);
```

7.2.2.1. Spatial Filtering

Spatial filtering is applied on five main stages in order to isolate RPE on a binary image. Table 21 describes main purposes of each spatial filtering stage.

Stage	Filtering Purposes
Image cut	- Use of function *CutImage* - Isolate useful image area
Red component isolation	- Remove green and blue components - Make use of strong RPE red component in order to isolate it - Normalize resulting 2D matrix in order to maximize contrast range
Blurring	- Apply a circular averaging filter of radius blur - Smooth rough noise on empty areas
Binary conversion	- Annul pixels lower then threshold - Differentiate interest region
Noise removal	- Apply a 2D median filter of size kernel - Remove salt & pepper like noise

Table 21 - Spatial Filtering Stages on RPE Detection

Figure 17 illustrates spatial filtering evolution on a sample OCT-SLO image.

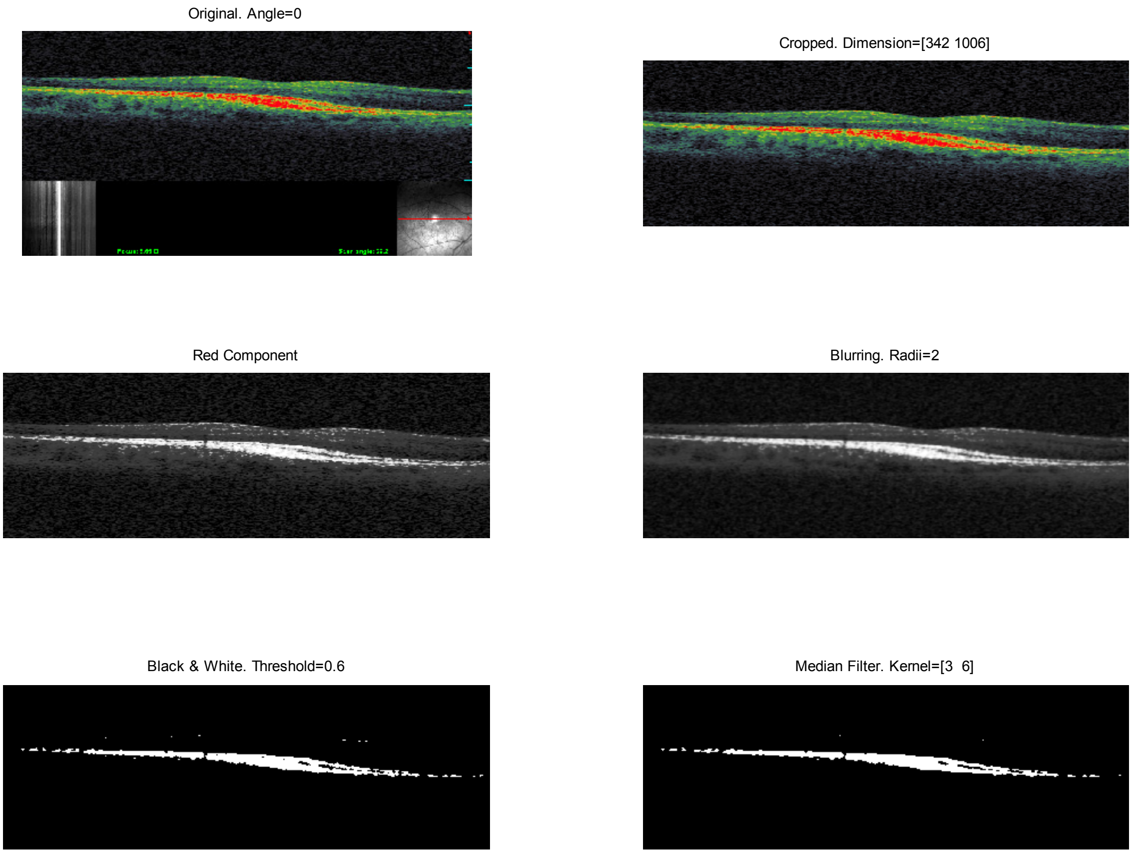


Figure 17 - Spatial Filtering Results on RPE Detection

7.2.2.2. Snake Filtering

At this part of the work it is intended to determinate the imaginary line that separates RPE from empty space above. Mentally, it is an easy task to perform, however in image processing this is a task that requires an unambiguous positioning of RPE.

Snake filtering is a method that progressively selects a neighbour pixel that matches scanning criteria. In this case, the purpose is to correctly detect a snake starting point and then horizontally scan binary image and select superior RPE edge points. Starting point can be placed anywhere in the image and horizontal scanning is performed on both directions.

Starting Point Detection

Detection of a starting point can be performed by horizontal and vertical scanning (in order to avoid remains of retinal edges) meaning that horizontal scanning allows finding both left and right starting points where as vertical scanning usually finds a near-center starting point.

Left starting point detection disregards several pixel columns, defined by borderRange (anticipates image border distortions), and includes various stages:

- Scan increment: iterative scan from left to right starting at borderRange index;
- Search for white pixels on current column: using method find, white pixels (value equals to 1) are searched. If no pixels are found, algorithm moves to next column;
- Lower pixel selection: when white pixels are found, the one with higher index (lower position) is selected, preventing outer RPE points selection;
- Top pixel detection: starting on lower pixel, a vertical scan is performed through white linked pixels. The superior pixel of this scanning is stored. This stage is performed by sub- function TopLinkedPoint;
- Setting of current column: current RPE vector position is set to detected top pixel. Boolean foundStart is made true.

Detection of right starting point is performed on a similar way. In this case, iterations direction is symmetric and the starting scan index is equal to image width less borderRange. Sub-function ScanHorizontal is applied to find both left and right starting points, as Figure 18 illustrates.

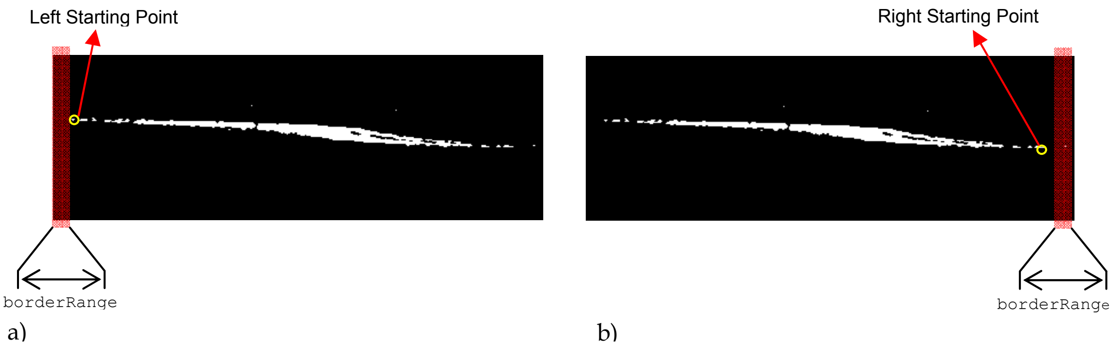


Figure 18 - Starting Points Achieved by Horizontal Scan on RPE Detection

Vertical scanning is performed by sub-function ScanVertical through several stages:

- Linear scan of the image: binary image lines are scanned from bottom to top in order to find white pixels;
- Selection of a relevant line: linear scanning stops when a line containing at least 5% of white pixels is detected. This way detection of noise or insignificant elements is prevented;
- Detection of all top pixels: for each white pixel of selected line, its higher linked point is found using sub-function TopLinkedPoint;
- Selection of starting point: from the set of points previously detected, highest one is stored on RPE vector as starting point. Selected column index is stored as the starting index for both left and right horizontal scanning methods.

Figure 19 illustrates described method. Selected line matches highlighted green line and blue rectangle shows analyzed pixels after line detection.

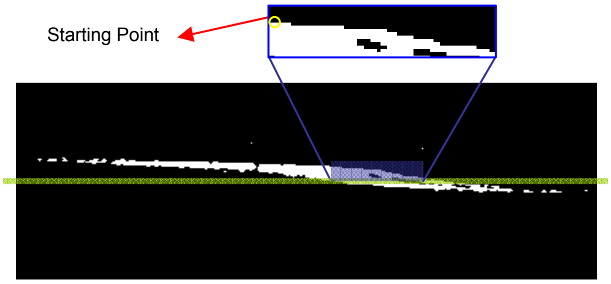


Figure 19 - Starting Point Selection by Vertical Scanning on RPE Detection

Horizontal Scanning

After starting point detection, similar snake filtering methods are applied on both `ScanHorizontal` and `ScanVertical` functions. `ScanHorizontal` function arguments are:

```
[epithelium equals] = ScanHorizontal(img_bit, increment, neighbor, meanRange,
                                     bottomScale, increaseNeighbor, borderRange);
```

Before iterative scan begins, starting and ending column indexes are set, depending on provided `increment` value. If `increment` is equal to 1 iterative scan occurs from `borderRange` to last column, otherwise if `increment` is equal to -1 iterative scan occurs from `width-borderRange` to first column. After starting point detection, iterative snake algorithm proceeds as it is described:

- Setting of search range: superior and inferior white pixels search limits on current column depend on the coordinates of last detected RPE point (`last`). According to Matlab axes system, current searching top limit is defined by `last-neighTemp` (by default, `neighTemp` is equal to `neighbor`). Bottom limit is equal to `last+neighTemp*bottomScale` (allows longer inferior limits);
- Search for white pixels: if white pixels are found on previously set interval their indexes are arranged to match image dimensions;
- Setting of candidate index: after white pixels search, it is necessary to determine which index has more chances of being stored in RPE vector. This process can occur on three distinct ways:
  - No white pixels found: current RPE vector index is equal to `last`. Search interval is increased by `increaseNeighbor` – Figure 20.a);
  - One white pixel found: found pixel is stored as a RPE vector candidate. The value of `neighTemp` is reset to `neighbor` – Figure 20.b);
  - Several white pixels found: function `TopLinkedPoint` is applied in order to select an index near to `last`, otherwise it is selected the superior found point - Figure 20.c). The value of `neighTemp` is reset to `neighbor`;

- Setting of index to save: in order to avoid hard transitions between contiguous RPE pixels, the maximum transition value between adjacent points is two units. Thus, all candidate indexes localized farther then or equal to two units from `last` are stored only two units away. This stage is performed by function `setCoordinate` and current `epithelium` index is saved.
- Scanning of the section beyond `borderRange`: when image scan terminates, ignored section outside `borderRange` is scanned on the opposite direction using `ScanHorizontalBorder`. Sub-function `ScanHorizontalBorder` performs steps equal to these described previously.

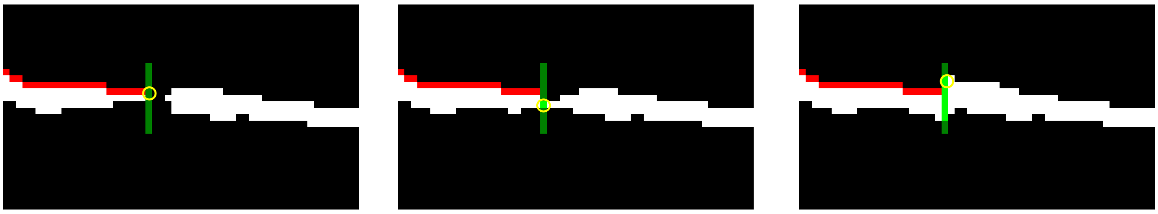

a) b) c)

Figure 20 - Illustration of a Left to Right Snake Scanning on RPE Detection
RPE is represented by white section.
Red line represents defined RPE vector indexes.
Green line illustrates current found white pixels.
Yellow circle indicates current RPE candidate pixel.
Scanning from right to left has similar behavior.

After RPE vector detection, a movable average (range is equal to `meanRange`) is applied in order to soften transitions between coordinates. Previously, `epithelium` indexes that match white pixels on binary image are counted; this score is called `equals`, being useful as a scan selection criterion.

`ScanVertical` function arguments are:

```
[epithelium equals] = ScanVertical(img_bit, neighbor, meanRange, bottomScale,
                                                        increaseNeighbor);
```

Despite this function name, vertical scanning is only performed to find a starting point for horizontal snake scanning (`startIndex`). Stages of horizontal scanning are described above.

The major difference between `ScanVertical` and `ScanHorizontal` is the algorithm applied on starting point detection. As previously described, the starting point for `ScanVertical` function can be located anywhere on the RPE (not necessarily on left or right borders). After starting point detection two horizontal scanning events are performed: from `startIndex+1` to last column and from `startIndex-1` to first column. Both scanning methods are described above.

Once `epithelium` detection is completed, a movable average is performed from `startIndex` to both directions and score `equals` is also computed.

This alternative scanning method was developed in order to avoid that image edge distortions (e.g. optic nerve) affect starting point detection.

7.2.2.3. Scanning Selection

As reported on previous section, each scanning algorithm returns two variables: RPE indexes vector (`epithelium`) and white pixels score (`equals`).

After several tests on other possible scanning selection criterion factors, one conclusion was achieved: in most cases that were tested, the highest value of `equals` represents the best RPE detection.

Figure 21 demonstrates a successful RPE detection on an OCT-SLO image (white line). In this case the best scanning was performed by function `ScanVertical`.

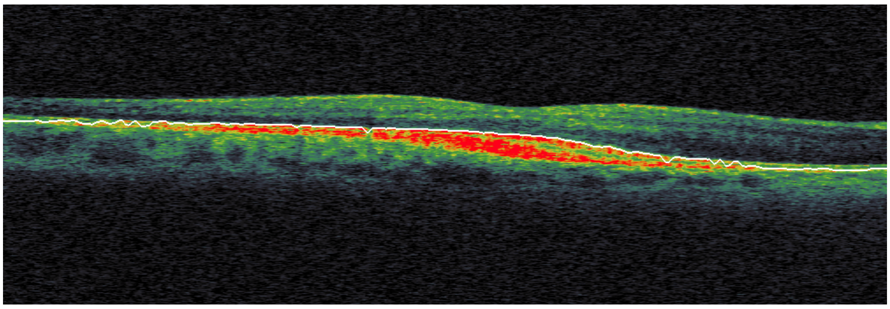


Figure 21 - RPE Detection on an OCT-SLO Image

7.2.3. Retinal Anterior Limit Detection

RAL detection algorithm has several characteristics in common with RPE detection algorithm. Thus, many applied methods will not be exhaustively described since they have already been explained on section 7.2.2.

This algorithm is based on three main methods that isolate retinal area from upper empty space: <u>spatial filtering</u>, <u>snake filtering</u> and <u>scanning selection.</u> These methods are performed by function `GetILM`. This function gets original image and several processing parameters (if parameters fields are empty, best found values are applied) and returns RAL coordinates matrix, detected hole coordinates and images from both filtering and scanning stages:

```
[ilm holes img_filtered img_display] = GetILM(display, display_final, img, filename,
kernel, threshold, neighbor, blur, holeDeep, minBorderWidth, borderRange, blackTopLimit)
```

7.2.3.1. Spatial Filtering

Spatial filtering is applied on five main stages (like in RPE detection) in order to isolate retinal area (all coloured region) from dark empty space (superior to retina). Table 22 describes main purposes of each spatial filtering stage.

Stage	Filtering Purposes
Image cut	- Use of function *CutImage* - Isolate useful image area
Greyscale conversion	- Remove blue component; use 70% red and 30% green components - Obtain a good compromise intensity/noise - Normalize resulting 2D matrix in order to maximize contrast range
Blurring	- Apply a circular averaging filter of radius blur - Smooth rough noise on empty areas
Binary conversion	- Annul pixels lower then column intensity average (or lower then threshold) - Differentiate interest region - Preserve relevant information on dark areas
Noise removal	- Apply a 2D median filter of size kernel - Remove salt & pepper like noise

Table 22 - Spatial Filtering Stages on RAL Detection

Figure 22 illustrates spatial filtering evolution on a sample OCT-SLO image. Entire coloured area is kept on binary transformation instead of keeping only hyper-reflective regions.

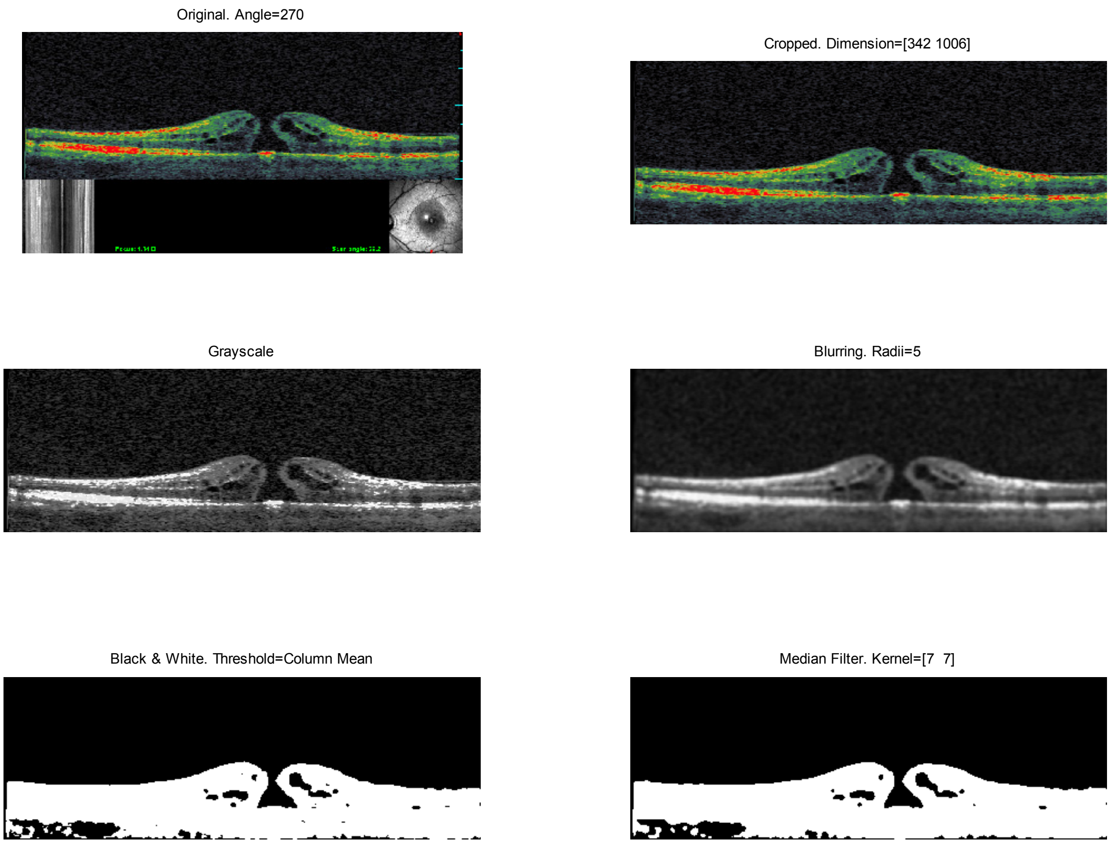


Figure 22 - Spatial Filtering Results on RAL Detection

7.2.3.2. Snake Filtering

At this stage it is intended to determinate the edge line that separates retina from empty space above. The detected line must be able to contour existing macular holes, which means that an index vector has not enough dimensions to store RAL info; thus, an index matrix (`ilm`) is required.

In this case, the purpose is to detect correctly a snake starting point and then scan binary image selecting retinal edge points.

Starting Point Detection

RAL detection is performed by scanning in both horizontal and vertical directions when a hole is detected. The starting point can be placed near left or right borders or it can be detected by vertical linear scanning from top to bottom.

Left starting point detection disregards several pixel columns, defined by `borderRange` and includes various stages (only stages that differ from previous section are described):

- Scan increment;
- Search for white pixels on current column;
- Higher pixel selection: when white pixels are found, the one with lower index (higher position) is selected, preventing lower RAL points selection;
- Pixel validation: selected pixel validation is performed by sub-function `FindStartEdge`. This function checks if there are at least `minBorderWidth` white neighbour pixels below current pixel (prevents from selecting noise dots or small elements above RAL);
- Setting of current column: LAR matrix is set to one at current pixel position (all preceding columns are set to one at the same position). Boolean `foundStart` is made true.

Detection of right starting point is performed on a similar way, as Figure 23 illustrates.

a) b)

Figure 23 - Left and Right Starting Points on RAL Detection

Starting point detection by vertical scanning is performed by function FullScanTopBottom through these stages:

- Location of top white pixels: indexes of the superior white pixel for each column are obtained using max method;
- Selection of top pixel: from detected superior pixels, the highest one is selected using min method. White pixels are searched on respective column;
- Pixel validation: sub-function FindStartEdgeOptimized is performed in order to check if there are at least minBorderWidth white neighbour pixels below current pixel. If true, current pixel is stored as startIndex.

Figure 24 illustrates described method.

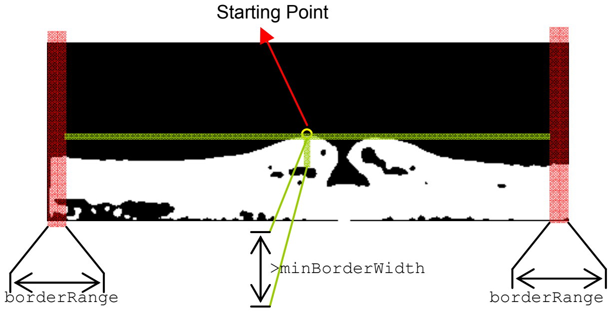


Figure 24 - Starting Point Selection by Vertical Scanning on RAL Detection

Horizontal Scanning

Horizontal scanning is performed by sub-functions FullScanLeftRight and FullScanRightLeft. These functions use similar processing methods with opposite directions.

```
[ilm edgeRatio holes] = FullScanLeftRight(img_bit, neighbor, increaseNeighbor,
                        borderRange, minBorderWidth, holeDeep, blackTopLimit);

[ilm edgeRatio holes] = FullScanRightLeft(img_bit, neighbor, increaseNeighbor,
                        borderRange, minBorderWidth, holeDeep, blackTopLimit);
```

A significant difference between these functions and scanning functions described on previous section is the dimension of stored indexes (on this case, edge indexes are saved on a 2D matrix – ilm – in order to store more then one index per column). Macular hole detection procedures caused significant changes on algorithms.

As it was described before, the first stage on these functions is to start point detection. Next stages can be described as follows:

- Setting of search range: superior and inferior white pixels search limits on current column depend on the coordinates of last detected RAL point (last). According to Matlab axes system, current searching top limit is defined by last-neighTemp and bottom limit is equal to last+holeDeep (defining maximum deep between adjacent pixels);

- Search for white pixels;
- Setting of indexes to save: if white pixels are found, sub-functions `TopLinkedPointMin` and `setCoordinate` (similar to sub-functions `TopLinkedPoint` and `setCoordinate`, described on previous section) are applied in order to get current RAL pixel (at maximum three units away from `last`). Sub-function `setCoordinate` returns the range of pixels (`interval`) that should be equalized to one on matrix `ilm`.
- Retinal hole scanning: a retinal hole is detected when no white pixels are found on search range (coordinate `startPoint` is stored). Several methods are required to perform hole edges detection, as it is described below and illustrated on Figure 25:
  - Top to bottom scanning: sub-function `ScanVerticalTopDown` allows to detect both sides of the retinal hole (input parameter `left` defines which side is being scanned), starting at `startPoint`. This function, similar to scanning functions described on previous section, stops edge detection process when the hole base is found (only white pixels are found on current search range), storing `vertLimit`;
  - Hole base scanning: this method, performed by sub-function `ScanHorizontal`, is similar to described scanning functions. This scan starts at `vertLimit` and stops when the opposite hole side is detected (only white pixels are found), storing `horizLimit`;
  - Bottom to top scanning: this stage (performed by sub-function `ScanVerticalDownTop`) starts at `horizLimit` and stops when no white pixels are found for `blackTopLimit` scanning iterations (retinal superior limit is achieved). Finally stopping coordinate (`topCoordinate`) is returned;
- LAR scanning: once retinal hole edges are detected, horizontal scanning proceeds from `topCoordinate`. Hole x-coordinates (`startPoint` and `topCoordinate` columns) are stored on `holes`.

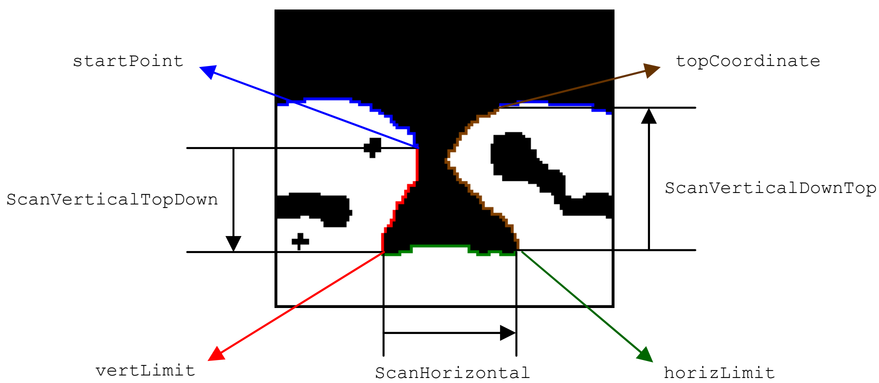


Figure 25 - Retinal Hole Detection

After RAL detection all edge pixels on `ilm` are counted by checking the existence of both white and black pixels around each RAL point. This score is divided by total number of detected points in order to get the ratio `edgeRatio`.

`FullScanTopBottom` function arguments are:

```
[ilm edgeRatio holes] = FullScanTopBottom(img_bit, neighbor, increaseNeighbor,
                        borderRange, minBorderWidth, holeDeep, blackTopLimit);
```

This function performs vertical scanning on starting point detection (`startIndex`) and then horizontal scanning is performed on both directions (sub-functions `ScanLeftRight` and `ScanRightLeft`) using methods similar to those described above. Also retinal holes are detected and rounded using these sub-functions (the score `edgeRatio` is again computed).

7.2.3.3. Scanning Selection

As reported on previous section, each scanning algorithm returns three variables: RAL indexes matrix (`ilm`), edge pixels score (`edgeRatio`) and retinal holes coordinates (`holes`). The best RLA detection method is the one that returns the highest `equalsFactor` (equals to `edgeRatio`).

Figure 26 demonstrates a successful RAL detection on an OCT-SLO image (blue line). In this case the best scanning was performed by function `FullScanRightLeft` (edgeRatio is equal to 1 because all detected points are edges) and one hole was detected.

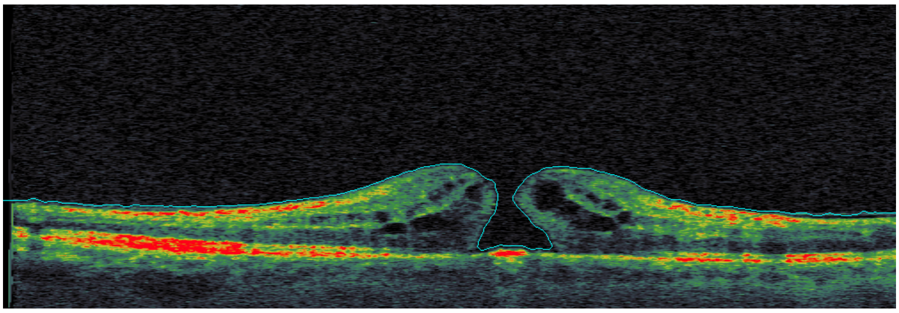


Figure 26 - RAL Detection on an OCT-SLO Image

7.3. Algorithm Tests

RPE detection algorithm was tested on a set of 735 valid images obtained on clinical environment. Selected images are representative of various types of retinal diseases and all anomalous images were removed. From this set, 549 are OCT-SLO images and 186 are

Stratus OCT images. All tests were separately analysed in order to validate RPE detection. Results are shown on Table 23.

	OCT-SLO	OCT	Total
Correct detection	527	170	697
Incorrect detection	22	16	38
Detection success (%)	95,993	91,398	94,830

Table 23 - RPE Detection Tests

RAL detection algorithm was tested on 545 valid OCT-SLO images (images showing distorted retinal limits were removed). All tests were separately analysed in order to validate RAL detection. Results are shown on Table 24.

	OCT-SLO
Correct detection	529
Incorrect detection	16
Detection success (%)	97,064

Table 24 - RAL Detection Tests

7.4. Software Integration

All developed image processing algorithms were implemented on patient management software module (class `ECRManagement_Images_Process()`). This is a temporary implementation since it is intended to include features detection on Data Mining module. Matlab integration on C# is provided by *Matlab Interface Library* [21].

Table 25 describes functionalities implemented on image processing GUI.

Features	Description	Matlab Function
Cut Image	Detect image format and recognize acquisition angle. Angle is shown at *Characteristics* field.	`CutImage`
RAL Detection	Delimit RAL and contour retinal macular holes.	`GetILM`
RPE Detection	Delimit RPE superior limit.	`GetEpithelium`
Retinal Serous Detachment Detection	Detects and contours existing retinal serous detachments.	`SerousDetachmentsDetect`
Detect and Fill Retinal Holes	Fills detected retinal holes.	`GetHoleInfo`
Edemas Detection	Detects and fills existing edemas.	`EdemasDetect`
Retinal Thickness Measure	Measures retinal thickness through macular region. Minimum, maximum and average thicknesses are shown at *Characteristics* field.	`GetRetinalThickness`
Mark Selected Components	Draws components selected on *Selected Features* check boxes.	`PaintImage`

Table 25 - Image Processing Features Implemented on Software Module

Figure 27 illustrates image processing window. The original OCT-SLO image is shown on the left top corner. User can mark detected features separately on processed image which is at the bottom.

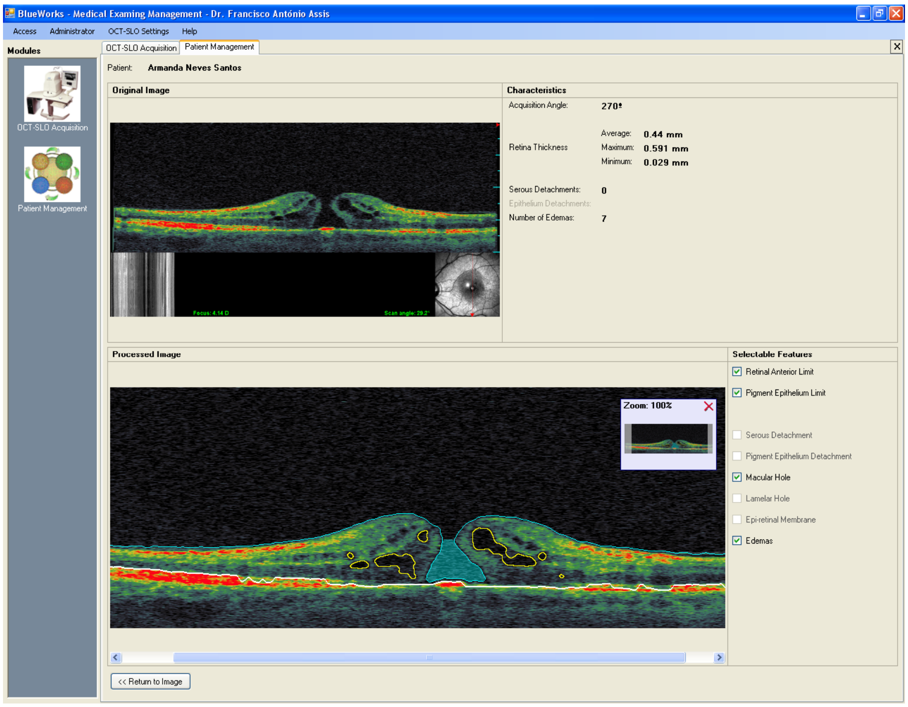


Figure 27 - Image Procesing Layout on Patient Management Module

8. Conclusions and Future Purposes

This chapter is divided in two sections. In the first section main project conclusions are discussed (emphasizing objectives, results and acquired skills). In the second section short term tasks are proposed in order to improve developed solution.

8.1. Conclusions

Once the ending of project academic component has been reached, the time to analyze and discuss developed work has come. At this moment, conclusions that describe both positive and negative facets can be introduced.

This project intends to create an innovator product, which caused a fuzzy definition of targets to reach at each work phase. The difficulty to schedule and to document tasks was one of the encountered obstacles. However, it was possible to reach an accord with project supervisors in order to create improved time tables.

The applied methodology did not always provide a logical sequence of tasks, namely to theoretical study component. However, CCC collaborators gave a crucial contribution on diseases and clinical process study stage. Dr. Travassos planning spirit was also very important due to his professional point of view.

When works at ISA started tasks planning became much more rigorous which allowed relieving resources. An equivalent partition of tasks to involved students was also an important step. A positive issue for the project was the integration with ISA's Software Development team. This integration allowed inspiring the conscientious spirit of a company that invests on research.

The study of CCC clinical process revealed a high dependence to paper and procedure replication, which generates higher costs and longer delays to patients. Developed solution will reduce required resources on both procedures. The study of eye diseases (namely retinal diseases) identifies numerous diagnosis subjects which are distinguishable by specific features.

The main remark that can be done to developed software component is that all proposed requisites are performed. In the current development stage, the solution does not respond to all constraints demonstrated on clinical process study. However the Acquisition Module will allow managing examinations on a more organized and dynamic way (when installed at CCC). Getting used to C# language was not a hard task, but some functionalities took much time to develop due to its specificity. Rules for good programming required at ISA, as well as error prevention mechanisms, were implemented. Another important characteristic is the system ability to be integrated on online functionalities.

Image processing component seriously increases the value of developed work because it illustrates one of these system great innovations: intelligent processing of medical data. Despite of the reduced number of detected features, positive result rates are very consistent, even with distorted images (low diagnosis relevance). Tests demonstrate that spatial filtering and snake algorithms are fast and efficient (when best-fitted researched parameters are provided). To implement this kind of algorithms it is necessary to think on a logical way (similar to the processor); on this subject, Matlab is an excellent tool because it allows working on an intuitive manner.

This project allowed the acquisition of several skills. First, surpassing stages gave ability to plan tasks at short term (or even at long term). The support of a multidisciplinary team to this project allowed wide exchange of views (especially on clinical matters) that were important for the improvement of performed tasks in all stages.

At programming level, the lack of Software Engineering experience was surpassed due to serious work and to support from ISA's software team. The ability to learn and research techniques for solving programming problems is another achieved purpose. In fact, a three step procedure seems like the best solving method: first, searching for existing solutions, then asking for other views and finally creating a specific solution.

The development of tasks using Matlab gave the opportunity to achieve new strategies for solving intelligent processing problems.

The project report was written in English language, which represents an extra challenge that provided new skills.

8.2. Future Purposes

Despite of their importance, tools developed in the course of this project did not achieve a product that could be sold. As predicted in the planning stage, the project is intended to have continuity. Thus, several purposes can be suggested in order to improve developed solution:

- Perform all tasks proposed on solution architecture (Diagram 6). Four main components can be consider in order to implement the full proposed system:
    - Creation and implementation of a Data Mining Module containing created algorithms and future intelligent processing methods. On this component, it is essential to test and improve new data mining methods (e.g. artificial neural networks and pattern recognition);
    - Integration of more examination equipments;
    - Creation of image processing algorithms for new types of examination images;
    - Connection to administrative software by creating a synchronization layer.

- Adjust and create image processing algorithms for detection of new features on OCT images. Several characteristics of OCT (or OCT-SLO) images are not yet detected with developed algorithms which means that new methods are required in order to perform a complete image scan.

- Create comparison techniques for applying in all examination images in order to discard incongruent feature detection.

- Improve Acquisition Module in order to automatically select relevant images. Thus, distorted and incongruent images should be discarded.

- Create, at short term, a software product to be placed on market (having the ability to integrate future modules).

- Provide, at long term, a similar solution over the Internet in order to allow remote consultations and examination analysis.

Thus, it is expected that all the work developed during this year provides a solid foundation to accomplish the main purposes as a mode to obtain a final product that should be able to change the way Medicine and Informatics are associated and therefore making the difference on diagnosis support.

References

[1] – Data Mining - http://www.statsoft.com/textbook/stdatmin.html

[2] - Santos, M. F.; Azevedo, C. (2005), *Data Mining – Descoberta de conhecimento em bases de dados*, FCA – Editora de Informática, Lisboa, ISBN 972 722 509 8

[3] - Valerio Netto, A. (2003), *Image processing and analysis for measuring ocular refraction errors*, Thesis (Doctorate), Institute of Mathematics and Computer Science, University of São Paulo, São Carlos – SP, Brazil

[4] - Fernández, D. C.; Salinas, H. M.; Puliafito, C. A. (2005), *Automated detection of retinal layer structures on optical coherence tomography images*, Bascom Palmer Eye Institute, University of Miami Miller School of Medicine, 1638 NW. 10th Ave, Miami, FL, *33136; USA*

[5] - Luis-García, R.; Martín-Fernández, M.; Arribas, J. I.; Alberola-López, C. (2003), *A fully automatic algorithm for contour detection of bones in hand*, Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática,Universidad de Valladolid, Valladolid, Spain

[6] - Souza, M. B.; Tomikawa, V. O.; Telles de Oliveira, B. F.; Polati, M. (2004),Neural network approach for planning surgical correction of strabismus, Departamento de Oftalmologia e Otorrinolaringologia, Faculdade de Medicina da Universidade de São Paulo, Brazil

[7] - CLASSE - Synthetic Classification for Industrial Supervision - http://www.cisuc.uc.pt/view_project.php?id_p=60

[8] - Human Eye - http://en.wikipedia.org/wiki/Eye

[9] - Dry Conjunctivitis - http://www.darkcirclesundertheeyes.com/dryeyesandsoreeyelids/eye-conditions-conjunctivitis.htm

[10] - Age-related Macular Degeneration - http://www.stlukeseye.com/Conditions/MacularDegeneration.asp

[11] - Retinal Detachment - http://www.theretinasource.com/conditions/retinal_detachment.htm

[12] - Retinal Pigment Epithelium Detachment - http://www.revoptom.com/HANDBOOK/SECT42a.HTM

[13] - Epiretinal membrane - http://www.retinaphysicians.com/vrs-conditions-epiretinalmembranes.htm

[14] - Fluorescein Angiography - http://en.wikipedia.org/wiki/Fluorescein_angiography

[15] - Heijl, A.; Patella, V. M. (2002), *Essential Perimetry, The Field Analyzer Primer, Third Edition*, Carl Zeiss Meditec AG, Jena, Germany, ISBN 0 9721560 0 3

[16] - Gupta, V.; Gupta, A.; Dogra, M. R. (2004), *ATLAS - Optical Coherence Tomography of Macular Diseases*, Taylor & Francis, London, ISBN 1 84184 468 3

[17] - Pierro, L. (2006), *Atlante di OCT/SLO - Una diversa chiave di lettura delle patologie retiniche*, Fabiano Editore, Canelli, Italy, ISBN 88 89629 19 3

[18] - C-Sharp - http://en.wikipedia.org/wiki/C_sharp

[19] - DICOM Parser for C# - Copyright © 2002 Fang Yang; http://www.sourceforge.net/projects/dicom-cs

[20] - Create Bitmaps - http://www.bobpowell.net/lockingbits.htm

[21] - Matlab Interface Library - Copyright © 2002 Emanuele Ruffaldi; http://www.sssup.it/

Appendices

Appendix 1 - Software Development and Image Processing Time Tables

Task	6-Feb	7-Feb	8-Feb	9-Feb	10-Feb	11-Feb	12-Feb	13-Feb	14-Feb	15-Feb	16-Feb	17-Feb	18-Feb	19-Feb	20-Feb	21-Feb	22-Feb	23-Feb	24-Feb	25-Feb	26-Feb	27-Feb	28-Feb
Definition of software and database structure	■																						
Familiarization with C# syntax and GUIs		■																					
Definition of GUI main structure		■																					
Create Examination Requests tab page			■																				
Create Select Images tab page			■				■																
Create Examination Report tab page			■				■																
Adaptation of Main Interface to plugin interface								■															
Implementation of GUI functionalities									■	■													
Implementation of DB access for basic functionalities										■				■	■	■							
Creation of 1st Intercalary Presentation																	■	■					
1st Intercalary Presentation																		■					
Setup of the new PC																						■	
Read DICOM files																						■	

Table 26 - February Time Table

Task	1-Mar	2-Mar	3-Mar	4-Mar	5-Mar	6-Mar	7-Mar	8-Mar	9-Mar	10-Mar	11-Mar	12-Mar	13-Mar	14-Mar	15-Mar	16-Mar	17-Mar	18-Mar	19-Mar	20-Mar	21-Mar	22-Mar	23-Mar	24-Mar	25-Mar	26-Mar	27-Mar	28-Mar	29-Mar	30-Mar	31-Mar
Read DICOM files	■																														
Get DICOM tags structure	■				■	■																									
Optimization of reading files method						■	■																								
Integration of DB queries						■	■	■				■	■	■	■																
Optimization of session flow and functionalities							■	■				■	■	■	■				■	■	■	■									
Get image from DICOM structure								■				■	■																		
Integration of DB insert and update methods													■	■	■				■	■	■	■				■					
Creation of calendar functionality																										■	■	■	■		
Creation of English language version																													■		

Table 27 - March Time Table

Task	1-Apr	2-Apr	3-Apr	4-Apr	5-Apr	6-Apr	7-Apr	8-Apr	9-Apr	10-Apr	11-Apr	12-Apr	13-Apr	14-Apr	15-Apr	16-Apr	17-Apr	18-Apr	19-Apr	20-Apr	21-Apr	22-Apr	23-Apr	24-Apr	25-Apr	26-Apr	27-Apr	28-Apr	29-Apr	30-Apr
Optimization of calendar functionality																														
Creation of administrator functionalities																														
Optimization of session flow on Main Interface																														
Optimization of English language version on MI and AM																														
Tests of application																														
Tests, conflict solving and optimization of DB methods																														

Table 28 - April Time Table

Task	1-May	2-May	3-May	4-May	5-May	6-May	7-May	8-May	9-May	10-May	11-May	12-May	13-May	14-May	15-May	16-May	17-May	18-May	19-May	20-May	21-May	22-May	23-May	24-May	25-May	26-May	27-May	28-May	29-May	30-May	31-May
MATLAB installation, gathering of CCC images																															
Creation of cut and angle detection algorithm																															
Creation of RPE detection algorithm																															
Integration of Matlab in patient module																															
Validation and optimization of RPE algorithm																															
Adaptation of RPE detection algorithm for RAL detection																															
Validation and optimization of RAL algorithm																															
Creation of 2nd Intercalary Presentation																															

Table 29 - May Time Table

Task	1-Jun	2-Jun	3-Jun	4-Jun	5-Jun	6-Jun	7-Jun	8-Jun	9-Jun	10-Jun	11-Jun	12-Jun	13-Jun	14-Jun	15-Jun	16-Jun	17-Jun	18-Jun	19-Jun	20-Jun	21-Jun	22-Jun	23-Jun	24-Jun	25-Jun	26-Jun	27-Jun	28-Jun	29-Jun	30-Jun	1-Jul	2-Jul
2nd Intercalary Presentation																																
Software and image processing documentation																																
Project Report																																

Table 30 - June Time Table

WIA-DM Project | Weekend / Holiday | CLASSE Project