



Technical University of Crete  
Electronics and Computer Engineering Department  
Electronics Division

# Hyper-Spectral excitation – emission microscopy

---

*Assembling and driving a tunable light source for spectroscopic microscopy*

Diploma Thesis by Odysseas Zografos

Thesis Committee:      Assoc. Professor Costas Balas (Supervisor)  
                                 Assist. Professor Matthias Bucher  
                                 Assist. Professor Yannis Papaefstathiou

August 2011, Chania

## Acknowledgements

I wish to express my sincere appreciation and thanks to Professor Costas Balas for his guidance and support throughout the implementation of this diploma thesis.

I am also grateful to Vassilis Kavvadias, MSc, George Epitropou, MSc, George Papoutsoglou, PhD candidate for their help and suggestions for the completion and improvement of the current work.

Also I would like to thank Professors Matthias Bucher and Yannis Papaefstathiou for their participation in the presentation and evaluation of my diploma thesis.

Last but not least, I wish to thank my friends, and especially Argyro Moraiti, for sharing my thoughts, worries and expectations during these years of my studies. I dedicate this thesis to my parents who supported me in every possible way throughout my studies.

## Abstract

In this diploma thesis, the development and evaluation of tunable light source is presented together with the integration procedure of this source with a spectral imager and a microscope. The integrated system is tested for spectroscopic microscopy of human tissues, which finds great application in medical histology. Also the configuration is tested for excitation/emission (fluorescence) microscopy, which is relatively new application but has an important diagnostic value.

While various spectral imaging devices and tunable light sources already exist and they are used in spectroscopic and fluorescence microscopy, the configuration present in this thesis is based on a linear variable band filter, which operates as monochromator. The system is capable of performing real-time spectral imaging in a plurality of spectral bands and micro-spectroscopy in any image pixel, in the spectral range 400-1000 nm. The wavelength-tuning step is 1 nm, while the full-width at half maximum in each step is about 4.1% of the operating central wavelength.

Lastly we developed a Graphical User Interface (GUI) that enables the user of the microscope to fully control and calibrate the whole configuration and acquire spectral data of the microscope's specimens in a variety of modes. Also the system integrated with a spectral imager was tested on microscope in order to acquire spectral data of cells and produce pseudo-color maps with diagnostic value.

# Contents

Introduction

Chapter 1 – Basic Principles and State-of-the-art Tunable Configurations

- 1.1 Electromagnetic Waves
- 1.2 Spectroscopy/Spectrometry
- 1.3 Sensor Types
- 1.4 Spectral Imaging
- 1.5 Excitation/Emission Microscopy
- 1.6 Tunable Light Sources
  - 1.6.1 Liquid Crystal Tunable Filter (LCTF)
  - 1.6.2 Acousto-optic Tunable Filter (AOTF)
  - 1.6.3 Optical Parametric Oscillator (OPO)

Chapter 2 – Hardware Configuration

- 2.1 White Light Source
- 2.2 Linear Variable Band Filter
- 2.3 Stepper Motor and the Driving Mechanism
  - 2.3.1 Connecting with the Serial Port
  - 2.3.2 Stepper Controller Breakdown
  - 2.3.3 Bipolar Stepper Motor
- 2.4 Microscopy Configuration
- 2.5 Spectral Camera

Chapter 3 – Software Design

- 3.1 Serial Communication with Stepper Motor
- 3.2 Console Application for Filter Movement
- 3.3 Sample GUI Application
- 3.4 FlyCap Embedded Application
- 3.5 Automated Calibration Operation

Chapter 4 – Overall System Performance and Evaluation

- 4.1 Accuracy of the Stepper Motor
- 4.2 Output Light Power and System Throughput
- 4.3 System Frame Rate

Chapter 5 – Application in Medical Histology

- 5.1 Sample Images of the Microscope Configuration
- 5.2 Spectroscopic Results
- 5.3 Excitation/Emission Results

Chapter 6 – Conclusions and Future Work

References

Appendix A

## Introduction

Hyper-Spectral imaging (HSI) (also imaging spectroscopy or chemical imaging) is the application of reflectance spectroscopy to every pixel in a spatial image. Spectroscopy can be used to detect individual absorption features due to specific chemical bonds in a solid, liquid, or gas, providing a unique spectral profile for identifying different materials. Actual detection is dependent on the spectral coverage, spectral resolution, and signal-to-noise ratio of the spectrometer, the abundance of the material and the strength of absorption features for the material in the wavelength region measured.

Spectral imaging finds many applications in analytical chemistry, bio-medicine and art work analysis. One of the bio-medicine applications is its use in spectroscopic microscopy, where spectral data are gathered from cell samples (or other microscope specimens) which usually are dyed with certain coloring substances. Another interesting region of application of spectral imaging is excitation/emission microscopy (or fluorescence microscopy), where an excitation illumination of the specimen changes its wavelength and a spectral imaging device (MuSIS HS) acquires the transmitted light through various filters. Fluorescence of substances is also very useful for detection and analysis of their properties. So in such kind of applications it is important to control the sample illumination according to the sample characteristics, meaning that the systems for these applications must have a certain amount of tunability.

In this thesis the design and development of a tunable light source are presented along with its integration with a spectral imager and a microscope. The system is analyzed in terms of hardware composition and driving as well as in terms of software design and construction. This hyper-spectral imaging system is tested and evaluated for spectroscopic and fluorescence microscopy. Although a lot of tunable light source systems exist [\[LINK\]](#), there is no configuration that can combine the excitation spectral range and high quality of spectral along with a cost-effective solution. The configuration presented in this thesis is low cost (comparing to other know devices) that can be used for spectroscopic and fluorescence microscopy very efficiently.

In **Chapter 1** the basic principles of spectral imaging and fluorescence are presented together with a state-of-the-art report about known tunable light sources configurations. **Chapter 2** analyzes the hardware's components and their function and gives important specifications of all the parts that are used in this configuration. **Chapter 3** analyzes all the driving, controlling and calibrating procedures that are encapsulated in the various versions of the software built to control the overall system. System's evaluation and performance are presented in **Chapter 4**. The results (acquired spectral data) of our hyper-spectral microscopy system are examined in **Chapter 5**. Lastly, **Chapter 6** discusses the advantages and drawbacks of our system and the improvements that can be made as future work.

# Chapter 1

## Basic Principles and State-of-the-art Tunable Configurations

Visual perception of scenes depends on appropriate illumination to visualize objects. The human visual system is limited to a very narrow portion of the spectrum of electromagnetic radiation, called light. In some cases natural sources, such as solar radiation, moonlight, lightning flashes, or bioluminescence, provide sufficient ambient light to navigate our environment. Because humankind was mainly restricted to daylight, fire was one of its first inventions as an artificial source of light.

Computer vision is not dependent upon visual radiation, fire, or glowing objects to illuminate scenes. As soon as imaging detector systems became available other types of radiation were used to probe scenes and objects of interest. Recent developments in imaging sensors cover almost the whole electromagnetic spectrum from x-rays to radio waves. In standard computer vision applications illumination is frequently taken as given and optimized to illuminate objects evenly with high contrast. Such setups are appropriate for object identification and geometric measurements. Radiation, however, can also be used to visualize quantitatively physical properties of objects by analyzing their interaction with radiation.

Physical quantities such as penetration depth or surface reflectivity are essential to probe the internal structures of objects, scene geometry, and surface-related properties. The properties of physical objects therefore can be encoded not only in the geometrical distribution of emitted radiation but also in the portion of radiation that is emitted, scattered, absorbed or reflected, and finally reaches the imaging system. Most of these processes are sensitive to certain wavelengths and additional information might be hidden in the spectral distribution of radiation. Using different types of radiation allows taking images from different depths or different object properties. As an example, infrared radiation of between 3 and 5  $\mu\text{m}$  is absorbed by human skin to a depth of  $< 1$  mm, while x-rays penetrate an entire body without major attenuation. Therefore, totally different properties of the human body (such as skin temperature as well as skeletal structures) can be revealed for medical diagnosis.

## 1.1 Electromagnetic Waves

Electromagnetic waves (Figure 1.1.1) were first postulated by James Clerk Maxwell and subsequently confirmed by Heinrich Hertz. Maxwell derived a wave form of the electric and magnetic equations, revealing the wave-like nature of electric and magnetic fields, and their symmetry. Because the speed of EM waves predicted by the wave equation coincided with the measured speed of light, Maxwell concluded that light itself is an EM wave. According to Maxwell's equations, a time-varying electric field generates a magnetic field and vice versa. Therefore, as an oscillating electric field generates an oscillating magnetic field, the magnetic field in turn generates an oscillating electric field, and so on. These oscillating fields together form an electromagnetic wave.

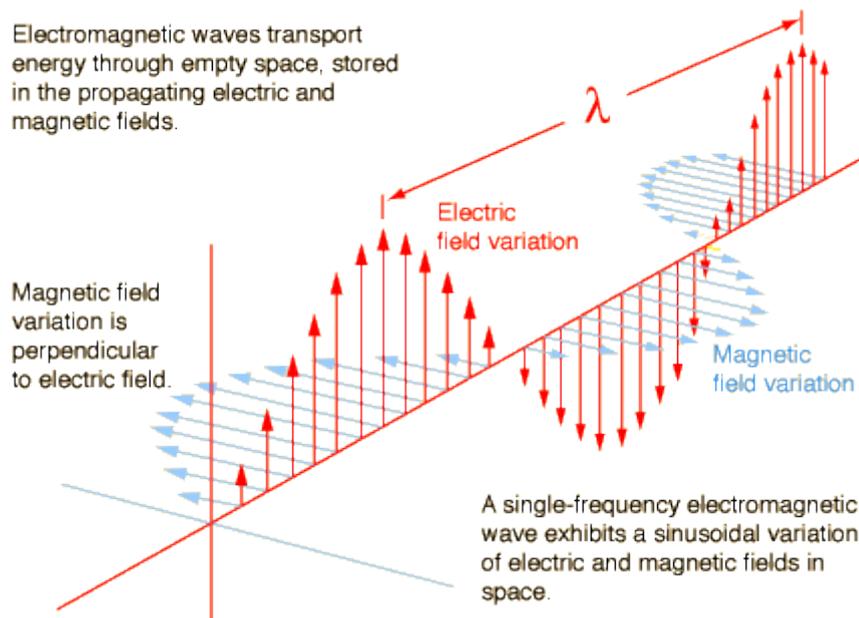


Figure 1.1.1 EM wave

In addition to electromagnetic theory, radiation can be treated as a flow of particles, discrete packets of energy called photons. One photon travels at the speed of light  $c$  and carries energy equal to:

$$E_{\text{photon}} = h \cdot \nu = \frac{h \cdot c}{\lambda},$$

where  $h = 6.626 \times 10^{-34} \text{ J s}$  is Planck's constant. Therefore the energy content of radiation is quantized and can only be a multiple of  $h\nu$  for a certain frequency  $\nu$ . While the energy per photon is given by  $E_{\text{photon}}$  the total energy of radiation is given by the number of photons. It was this quantization of radiation that gave birth to the theory of quantum mechanics at the beginning of the twentieth century.

Although photons do not carry electrical charge this unit is useful in radiometry, as electromagnetic radiation is usually detected by interaction of radiation with electrical charges in sensors. In solid-state sensors, for example, the energy of absorbed photons is used to lift electrons from the valence band into the conduction band of a semiconductor. The bandgap energy  $E_g$  defines the minimum photon energy required for this process. As a rule of thumb the detector material is sensitive to radiation with energies  $E_{\nu} > E_g$ .

Electric and magnetic fields obey the properties of superposition, so fields due to particular particles or time-varying electric or magnetic fields contribute to the fields due to other causes. (As these fields are vector fields, all magnetic and electric field vectors add together according to vector addition.) These properties cause various phenomena including refraction and diffraction. For instance, a travelling EM wave incident on an atomic structure induces oscillation in the atoms, thereby causing them to emit their own EM waves. These emissions then alter the impinging wave through interference.

Since light is an oscillation, it is not affected by travelling through static electric or magnetic fields in a linear medium such as a vacuum. In nonlinear media such as some crystals, however, interactions can occur between light and static electric and magnetic fields - these interactions include the Faraday Effect and the Kerr Effect. In refraction, a wave crossing from one medium to another of different density alters its speed and direction upon entering the new medium. The ratio of the refractive indices of the media determines the degree of refraction, and is summarized by Snell's law. Light disperses into a visible spectrum as light is shone through a prism because of the wavelength dependant refractive index of the prism material (Dispersion).

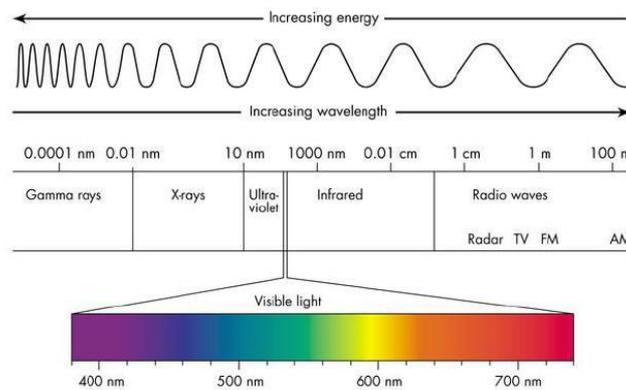


Figure 1.1.2 Spectrum of electromagnetic (EM) radiation

Monochromatic radiation consists of only one frequency and wavelength. The distribution of radiation over the range of possible wavelengths is called spectrum or spectral distribution. Figure 1.1.2 shows the spectrum of electromagnetic radiation together with the standardized terminology separating different parts. Electromagnetic radiation covers the whole range from very high energy cosmic rays with wavelengths in the order of  $10^{-16}$  m to sound frequencies above wavelengths of 106 m. Only a very narrow band of radiation between 380nm and 780nm is visible to the human eye. Each portion of the electromagnetic spectrum obeys the same principal physical laws. Radiation of different wavelengths, however, appears to have different properties in terms of interaction with matter and observability that can be used for wavelength selective detectors.

## 1.2 Spectroscopy / Spectrometry

Spectroscopy was originally the study of the interaction between radiation and matter as a function of wavelength ( $\lambda$ ). In fact, historically, spectroscopy referred to the use of visible light dispersed according to its wavelength, e.g. by a prism. Later the concept was expanded greatly to comprise any measurement of a quantity as function of either wavelength or frequency. Thus it also can refer to interactions with particle radiation or to a response to an alternating field or varying frequency ( $\nu$ ). A further extension of the scope of the definition added energy ( $E$ ) as a variable, once the very close relationship  $E=h\nu$  for photons was realized. Spectrometry is the spectroscopic technique used to

assess the concentration or amount of a given species. In those cases, the instrument that performs such measurements is a spectrometer or spectrograph. Spectroscopy is often used in physical and analytical chemistry for the identification of substances through the spectrum emitted from or absorbed by them. Spectroscopy/spectrometry is also heavily used in astronomy and remote sensing. Most large telescopes have spectrometers, which are used either to measure the chemical composition and physical properties of astronomical objects or to measure their velocities from the Doppler shift of their spectral lines.

### 1.1.1 Nature of excitation measured

The type of spectroscopy depends on the physical quantity measured. Normally, the quantity that is measured is intensity, either of energy absorbed or produced.

**Electromagnetic spectroscopy** involves interactions of matter with electromagnetic radiation, such as light.

*Electron spectroscopy* involves interactions with electron beams. Auger spectroscopy involves inducing the Auger effect with an electron beam. In this case the measurement typically involves the kinetic energy of the electron as variable.

*Mass spectrometry* involves the interaction of charged species with magnetic and/or electric fields, giving rise to a mass spectrum. The term "mass spectroscopy" is deprecated, for the technique is primarily a form of measurement, though it does produce a spectrum for observation. This spectrum has the mass  $m$  as variable, but the measurement is essentially one of the kinetic energy of the particle.

*Acoustic spectroscopy* involves the frequency of sound.

*Dielectric spectroscopy* involves the frequency of an external electrical field

*Mechanical spectroscopy* involves the frequency of an external mechanical stress, e.g. a torsion applied to a piece of material.

### 1.1.2 Measurement process

Most spectroscopic methods are differentiated as either atomic or molecular based on whether or not they apply to atoms or molecules. Along with that distinction, they can be classified on the nature of their interaction:

*Absorption spectroscopy* uses the range of the electromagnetic spectra in which a substance absorbs. This includes atomic absorption spectroscopy and various molecular techniques, such as infrared spectroscopy in that region and nuclear magnetic resonance (NMR) spectroscopy in the radio region.

*Reflectance/Emission spectroscopy* uses the range of electromagnetic spectra in which a substance radiates (emits). The substance first must absorb energy. This energy can be from a variety of sources, which determines the name of the subsequent emission, like luminescence. Molecular luminescence techniques include spectrofluorimetry.

*Scattering spectroscopy* measures the amount of light that a substance scatters at certain wavelengths, incident angles, and polarization angles. The scattering process is much faster than the absorption/emission process. One of the most useful applications of light scattering spectroscopy is Raman spectroscopy.

### 1.3 Sensor Types

We define a sensor as an instrument capable of measuring electromagnetic radiation. There are hand held sensors (e.g., field spectrometer) that can be used in the laboratory or during field work for reference measurements. Multispectral sensors contain multiple detectors that are sensitive to specific ranges of the EM spectrum (spectral bands). Typically, multispectral sensors have a few bands ( $< 10$ ), super-spectral sensors have many ( $> 10$ ) bands, and hyper-spectral sensors have up to a few hundred spectral bands.

Another classification scheme of sensors concerns scanning: whiskbroom and pushbroom scanners. Their introduction is important to understand some of the causes of errors in the image data we have to deal with in data processing.

The whiskbroom scanner uses a rotating mirror to scan the target surface. It directs a narrow beam of energy onto the detector. An important factor of the whiskbroom is the instantaneous field of view (IFOV) of the scanner. It is the field of view (or cone angle) of the mirror at the instant that the energy is sensed on the detector. Because there is only one detector for each spectral image, we do not have to deal with inter-calibration problems between neighboring pixels. As a drawback, there is the mechanical part of the rotating mirror, limiting resolution due to slower data rates. 15

The pushbroom scanner uses a wide-angle optical system that focuses on a strip across the whole of the scene onto a linear array of CCD (Charged Coupled Device) or CMOS (Complementary Metal Oxide Semi-conductor) detectors. The signal from each detector is sampled to create a record for the across track pixels. Each column of each spectral image is acquired with a different sensor, which can lead to striping artifacts. On the other hand, high data rates can be obtained, which make this technique popular for high resolution sensors.

Because sensors are a type of transducer, they change one form of energy into another. For this reason, sensors can be classified according to the type of energy transfer that they detect: thermal sensors, heat sensors, electromagnetic sensors, metal, chemical, optical radiation sensors.

Optical radiation sensors can be further classified:

- Light time-of-flight. Used in modern surveying equipment, a short pulse of light is emitted and returned by a retro-reflector. The return time of the pulse is proportional to the distance and is related to atmospheric density in a predictable way.
- **Light sensors, or photo-detectors**, including semiconductor devices such as photocells, photodiodes, phototransistors, CCDs, and Image sensors; vacuum tube devices like photo-electric tubes, photomultiplier tubes and mechanical instruments such as the Nichols radiometer. Solid-state detectors detect light by causing photons to excite electrons from immobile, bound states of the semiconductor (the valence band) to a state where the electrons are mobile (the conduction band). The mobile electrons in the conduction band and the vacancies, or —holes, in the valence band can be moved through the solid with externally applied electric fields, collected onto a metal electrode, and sensed as a photo-induced current.

Micro-fabrication techniques developed for the integrated-circuit semiconductor industry are used to construct large arrays of individual photodiodes closely spaced together. The device, called a charge-coupled device (CCD), permits the charges that are collected by the individual diodes to be read out separately and displayed as an image.

- Infra-red sensor, especially used as occupancy sensor for lighting and environmental controls.
- Proximity sensor- A type of distance sensor but less sophisticated. Only detects a specific proximity. May be optical - combination of a photocell and LED or laser. Applications in cell phones, paper detector in photocopiers, auto power standby/shutdown mode in notebooks and other devices. May employ a magnet and a Hall Effect device.
- Scanning laser- A narrow beam of laser light is scanned over the scene by a mirror. A photocell sensor located at an offset responds when the beam is reflected from an object to the sensor, whence the distance is calculated by triangulation.
- Focus. A large aperture lens may be focused by a servo system. The distance to an in-focus scene element may be determined by the lens setting.
- Binocular. Two images gathered on a known baseline are brought into coincidence by a system of mirrors and prisms. The adjustment is used to determine distance. Used in some cameras (called range-finder cameras) and on a larger scale in early battleship range-finders
- Interferometry. Interference fringes between transmitted and reflected light waves produced by a coherent source such as a laser are counted and the distance is calculated. Capable of extremely high precision.

## 1.4 Spectral Imaging

Spectral imaging (SI) combines spectroscopy with imaging. A spectral imager provides spectral information at each pixel of a 2 dimensional detector array. The SI systems acquire a 3-dimensional data set of spectral and spatial information, known as **spectral cube**. The spectral cube can be considered as a stack of images, each of them acquired at a different wavelength. Imaging spectroscopy can be considered as the equivalent of color photography, but each pixel needs to acquire many bands of light intensity data from the spectrum, instead of just the three bands of the RGB color model. More precisely, it is the simultaneous acquisition of spatially co-registered images in many spectrally contiguous bands (Figure 1.4.1). Combined spatial and spectral information offers great potential for the non-destructive/invasive investigation of a variety of studied samples.

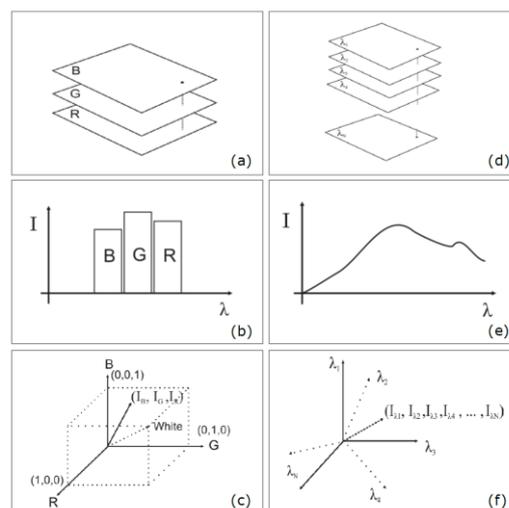


Figure 1.4.1 Image data capturing and representation in color (a, b, c) and spectral (d, e, f) cameras

Different spectroscopy types and modalities exist, depending on the optical property that it is intended to be measured, namely: absorption, spontaneous emission (fluorescence, phosphorescence), scattering (Rayleigh elastic, Raman inelastic) spectroscopy etc. As the light travels into the sample, photons are experiencing absorption, which may result in fluorescence emission and multiple scattering due to the local variation of the index of refraction. Spectrometers measure the intensity of the light emerging from the sample as a function of the wavelength. The collected light passes through a light dispersing element (grating), which spatially splits the light wavelengths onto the surface of an optical sensor array, interfaced with a computer for recording and processing the spectrum. Sample illumination can be provided by either a broad band (e.g. white light) or a narrow band light source. In the first case, the measured spectra provide information for the absorption and scattering characteristics of the tissue. In the second case, the measured spectra probe the fluorescence characteristics of the sample. A sensitive optical sensor is used for collecting the emission spectra.

The collected emission spectra can provide diagnostic information for the compositional status of the sample. This makes spectroscopy an indispensable tool for non-destructive analysis and for the development of novel, non-invasive diagnostic approaches. In biomedical sciences particularly, the diagnostic potential of tissue spectroscopy is based on the assumption that the absorption, fluorescence and scattering characteristics of the tissue change during the progress of the disease. Over the last 20 years spectroscopy has been extensively investigated as a tool for identifying various pathologic conditions on the basis of their spectral signatures. It has been demonstrated that spectroscopy can successfully probe intrinsic or extrinsic chromophores (molecules that are able to absorb light) and fluorophores (molecules that are able to absorb and emit light), the concentration of which change during the development of the disease. In its conventional configuration, spectroscopy uses single point probes that cannot easily sample large areas or small areas at high spatial resolution. It is obvious that this configuration is clearly suboptimal when solid and highly heterogeneous materials, such as the biological tissues, are examined. In these cases, the collected spectrum is the result of the integration of the light emitted from a great number of area points. This has the effect of mixing together signals originating from both pathologic and healthy areas, which makes the spectral signature-based identification problematic. Looking at the same problem from another perspective, point spectroscopy is considered as inefficient in cases where the mapping of some characteristic, spectrally identifiable property, is of the utmost importance.

Spectroscopy probes optical signals with high spectral resolution but with poor spatial resolution. The vastly improved computational power together with the recent technological developments in tunable optical filter and imaging sensor technologies have become the catalysts for merging together imaging and spectroscopy. Both areas – imaging and spectroscopy- continue to be impacted by technological innovations that enable faster acquisition of superior-quality data. SI has the unique feature of combining the advantages of both imaging and spectroscopy (high spatial and spectral resolution) in a single instrument. In SI, light intensity is recorded as a function of both wavelength and location. In the image domain, the data set includes a full image at each individual wavelength. In the spectroscopy domain, a fully resolved spectrum at each individual pixel can be recorded. These devices can measure the spectral content of light energy at every point in an image. Multiple images of the same scene at different wavelengths are acquired for obtaining the spectra. As an example, a SI device integrating an imaging sensor with 1000X1000 pixels provides 1 million individual spectra. A spectrum containing 100 data points results from an equal number of spectral images. Assuming that the intensity in each pixel is sampled at 8bit then the size of the resulting spectral cube equals to 100 Mbytes. Due to the huge size of the collected datasets, SI data processing, analysis and storage require fast computers and huge mass memory devices.

SI can be easily adapted to a variety of optical imaging instruments such as camera lenses, telescopes, microscopes, endoscopes etc. For this reason applications of SI span from planet and earth inspection (remote sensing) to internal medicine and molecular biology. Particularly, in clinical diagnosis spectral mapping can provide information for the topography, the severity and the margins of the lesion for diagnosis and for guiding treatment.

## 1.5 Excitation/Emission Microscopy

Spectral Imaging finds great application in light microscopy [\[LINK\\_pap\]](#), since an SI apparatus provides the capability to acquire process and compare specimens or regions of a specimen based on their spectral characteristics. In a common SI configuration used for spectroscopic microscopy there is only one filtering component (used as monochromator). In order to acquire images containing information of one spectral band each time (and then process them as one spectral cube) it is essential either to filter the excitation light or filter the emitted/transmitted light from/through the microscope's specimen. This way the light inserted the camera is monochromatic (or with a narrow spectral bandwidth), and thus the images acquired present the light-interaction properties of the specimen in the specific wavelength.

Excitation/emission microscopy configurations are based in the existence of two filtering components; one placed before the specimen that filters the white/broadband light of the illuminating source to a monochromatic/narrow-band excitation radiation, and one placed after the specimen to filter the emitted/transmitted light before entering the spectral camera. Controlling both the excitation and the emission wavelengths in a microscope configuration triggers two very useful capabilities. The first is the capability to analytically scan and observe the bandwidth of the excitation light as well as to discover in which bands the specimen produces the images with best contrast and most diagnostic value. This capability allows the researcher to select several excitation and emission wavelengths that combine for the most clear and valuable spectral cubes.

The second capability and the most important one is that a excitation/emission microscopy configuration as described above enables fluorescence microscopy.

### 1.5.1 Principles of Fluorescence

Fluorescence is defined as a luminescence which occurs essentially only during the irradiation of a substance by electromagnetic radiation [\[LINK\\_IUPAC\]](#). More simply can be defined as the molecular absorption of light energy (photon) at one wavelength and its re-emission at another, usually longer, wavelength. The fluorescence process can be broken down into three parts, which are all presented in Figure 1.5.1.1.

- **Excitation** - absorption of light of an appropriate wavelength by fluorophore (Figure 1.5.1.1a)
- **Excited state** - fluorophore undergoes vibrational and conformational changes (Figure 1.5.1.1b)

- **Emission** - photon of light is emitted (Figure 1.5.1.1c), this transition which generates a photon of different wavelength is known as Stokes' shift.

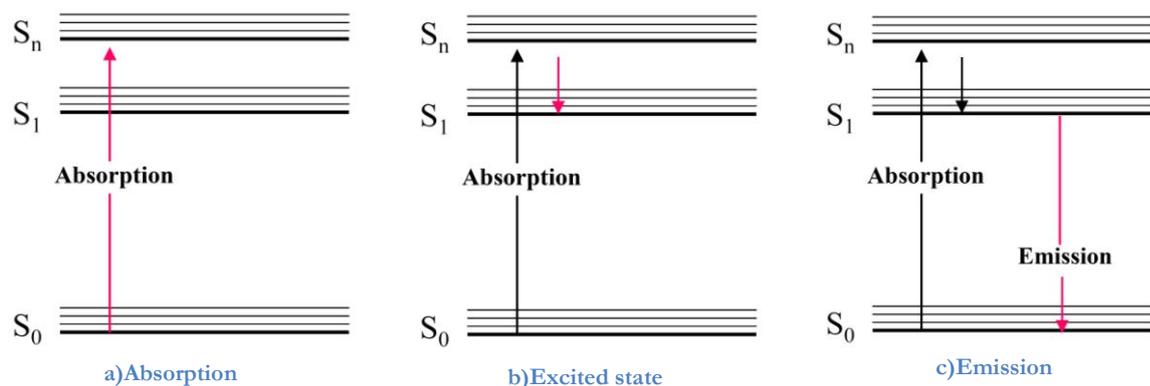


Figure 1.5.1.1 Parts of fluorescence process

Early investigations showed that many specimens (minerals, crystals, resins, crude drugs, butter, chlorophyll, vitamins, inorganic compounds, etc.) fluoresce when irradiated with UV light. In the 1930s, the use of fluorochromes began in biology to stain tissue components, bacteria, or other pathogens. Some of these stains were highly specific and they stimulated the development of the fluorescence microscope.

### 1.5.2 Fluorescence Microscopy

Fluorescence microscopy has become an essential tool in biology as well as in materials science as it has attributes that are not readily available in other optical microscopy techniques. The use of an array of fluorochromes has made it possible to identify cells and submicroscopic cellular components and entities with a high degree of specificity amid nonfluorescing material. The fluorescence microscope can reveal the presence of a single fluorescing molecule. In a sample, through the use of multiple staining, different probes can simultaneously identify several target molecules. Although the fluorescence microscope cannot provide spatial resolution below the diffraction limit of the respective objects, the detection of fluorescing molecules below such limits is readily achieved.

## 1.6 Tunable Light Sources

In this paragraph some of the most commonly used tunable light sources configurations is presented and discussed. This section of the thesis is important in order to compare the system described in the next chapters with some known devices of the same gender.

### 1.6.1 Liquid Crystal Tunable Filter (LCTF)

A liquid crystal tunable filter (LCTF) is based on a Lyot filter, invented by Bernard Lyot in 1920. The Lyot filter employs polarizing interferometry to yield a narrow-band filter suitable for recording monochromatic images. It consists of  $N$  birefringent plates (retarders) each positioned between parallel polarizers oriented such that the polarizer axis forms an angle of 45 degrees to the optic axis.

As it is known, a birefringent crystal, such as calcite or boron nitride, exhibits two different indices of refraction. Because of that, linearly polarized light propagating through the crystal is resolved into orthogonally polarized components, called “ordinary” and “extraordinary” rays, traveling at different speeds. This introduces a mutual optical path difference ( $d\Delta n$ ) between these two components, where  $\Delta n = n_o - n_e$  is the birefringence,  $n_o$  and  $n_e$  are the ordinary and extraordinary refractive indices and  $d$  the thickness of the crystal. The associated mutual phase retardation is then  $\Gamma = 2\pi d\Delta n / \lambda$ , where  $\lambda$  is the wavelength. The 45° angle between the polarizer and optic axes indicates that equal ordinary and extraordinary components are transmitted. At the crystal’s exit these two components are superimposed and their combined polarization depends on their mutual phase retardation. Phase retardation can turn the polarization plan of the light transmitted through the crystal so that it can in general form a non-zero angle with the axis of the output polarizer (analyzer). The projected to the analyzer’s axis light intensity component will be finally transmitted from the polarizer-retarder-analyzer stage. The transmitted intensity or the transfer function of the stage is given by the formula:

$$T = \cos^2(\Gamma/2) = \cos^2(\pi d\Delta n / \lambda) \text{ (eq. 1)}$$

As in can be seen in equation (1), the light transmittance of the stage is a function of  $d$ ,  $\Delta n$ , and  $\lambda$ . Assuming a birefringent crystal with a given  $d$  and  $\Delta n$ , and broad band light entering the stage, a certain set of wavelengths will be transmitted, while the remaining wavelengths will be rejected as not fulfilling the transmission criteria. In order to reduce the bandwidth and the number of the transmitted spectral bands, several cascaded polarizer-retarder-analyzer stages are assembled in series, with the output of one being the input of the other (Figure 1.6.1.1).

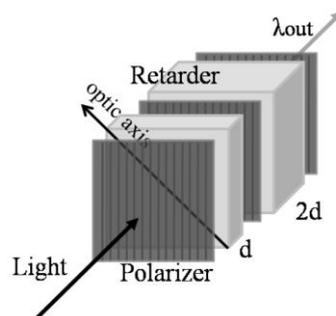


Figure 1.6.1.1 A two stage Lyot filter assembly composed of polarizers and retarding elements, where  $d$  is the thickness of the first retarder and  $\lambda_{out}$  is the transmitted wavelength.

The thickness of the retarder varies from stage to stage and more specifically, the ratio between the thicknesses of consecutive retarders (waveplates) is a factor of two. Particularly, if the thickness of the thinnest plate is  $d$  then the thickness of the  $j$ th plate is  $2^{j-1}d$ . The spectral transmission function of an  $N$ -waveplate Lyot filter is the product of the transmission functions of all constituent waveplate/polariser assembly and is given by

$$T = \prod_{j=1}^N \cos^2(2^{j-2} \Gamma) \quad (\text{eq. 2})$$

Apart from the optimum number of stages determining the spectral discrimination performance of the Lyot filter, it can be proved that the thickness of the first stage determines the free spectral range and the thickness of the final stage determines the spectral resolving power (Brady 2009). What it has been described so far is a multistage Lyot filter, which with a given set of alternating birefringent retarders and analyzers it behaves as a bandpass filter, transmitting certain spectral band(s). Tunability of the transmitted bands is achieved by modifying the Lyot filter for enabling the external control of the birefringence and hence the control of phase retardation (eq. 2). The Lyot filter becomes a liquid crystal tunable filter (LCTF) when thin liquid crystal layers, acting as electronically controlled phase retarders, are added to each stage side-by-side with the fixed retarders.

Two transparent electrodes are placed on either side of the liquid crystal waveplate. The liquid crystal cell contains nematic crystals that are aligned with their long axis nearly perpendicular to the light path. The inner surfaces of the liquid crystal cell are prepared in such a way that the molecules have a preferred orientation parallel to the surface. However, when a voltage is applied across the electrodes an electric field parallel to the light path generates a torque that twists the liquid crystals into the direction of electric field. The molecules are then aligning with the applied electric field, thus decreasing the retardance through the liquid crystal waveplate (Slawson 1999). The electronically adjustable retardance enables the tuning of the transmitted wavelength. The LCTF's spectral transmittance depends on the phase shift, which inherently depends on the total retardance by both the fixed and the liquid crystal waveplate adjustable retarders. LCTF is a wonderful device offering versatile and relatively fast wavelength selection, without much image distortion or shift. Switching speed is limited by relaxation time of the liquid crystals, being of the order of  $\sim 50$  msec. Special devices can be designed for fast switching ( $\sim 5$  msec) through a short sequence of wavelengths. LCTFs can cover both visible and infrared spectral bands but with different module assemblies, each spanning approximately one octave of wavelength (e.g. 400-750nm). The minimum output bandwidth is 5nm increasing with the wavelength up to 30nm or more. The major problem of LCTFs is their poor light throughput since half of the light corresponding to one polarization state is rejected by the input polarizer, and peak transmission of the other half probably is less than 40%. Poor filter transmittance will require long sample and sensor exposure times especially in low-light applications. Long light exposure of the sample would result in unwanted effects such as fluorophore photobleaching. Long camera integration times would in practice cancel the fast switching ability of a spectral imager integrating an LCTF. Insufficient out-of-band rejection is a second problem of LCTFs requiring extra optics for blocking light leaks. Finally, due to the thermal instability of the liquid crystals maximum operating temperature of LCTFs is low, ranging between 10o-40o, which should be taken into account when the filter is intended to be used for filtering light sources.

LCTF's science and technology evolves very rapidly because of its proven applicability in a variety of biomedical imaging applications. Recently, a number of more efficient, compact and less expensive liquid crystal based tunable filter configurations have been developed, departing from the Lyot's filter concept. Amongst the most interesting implementations, are the Holographic polymer-dispersed liquid crystals (H-PDLC) (Woltman 2007, Bowley 2001, Qi 2004, Fox 2007) and the vertically aligned

deformed helix ferroelectric liquid-crystals (VA-DHFLC) (Woltman 2007, McMurdy 2006). H-PDLCs and VA-DHFLCs form photonic crystals with a tunable photonic band gap determining a range of wavelengths, which are reflected by the crystal, according to the coupled mode theory. In each case, the optical anisotropy and the responsivity of liquid crystals to electric field or heating is exploited for altering or deactivating the wavelength selectivity of the photonic crystal. The spectral resolution of both the H-PDLC and the VA-DHFLC is in the range of 15-20nm. Another very exciting progress in liquid crystal tunable filter science and technology is the liquid crystal tunable Fabry-Perot filters (LCFP). Fabry-Perot cavities are optical resonators consisted of two planar, partially transparent, reflective surfaces, separated by an intermediate medium which can be air or some kind of dielectric material. A Fabry-Perot filter is penetrable by wavelengths for which a standing wave condition is satisfied inside the cavity. In a LCFP, a liquid crystal is used as the medium. By applying potential to the crystal, the refractive index of the medium changes and therefore the transmission modes supported by the cavity are affected. This enables the control of the transmitted wavelength. Using this configuration, switching times of the order of few milliseconds can be achieved. The minimum bandwidth in each tuning step is very narrow (0.05nm) but the tuning range is not larger than 100nm. A succession of cavities may be used for broadening the range of frequencies but such a configuration is inefficient in terms of throughput, being at the same time more expensive.

LCTFs have been proved as an indispensable tool in biomedical optical imaging in wide range of *in vivo* and *in vitro* applications. Attached to an endoscope, LCTFs were proved efficient in detecting small differences in fluorescent properties of malignant versus non-malignant mice tissues (Martin 2006). Integrated with a colposcope, the LCTF was used to identify the optimum spectral bands for monitoring the uptake kinetics of biomarkers for improving diagnostic accuracy of cervical neoplasia (Balas 2001). LCTFs have also been used in hemodynamic imaging studies, in mapping of oxygen transfer and oxygen saturation levels in tumors (Sorg 2005, Scala 2009). In this study, the variation of the oxy- and deoxy-hemoglobin optical signatures in the red/NIR range of wavelengths is investigated and the information obtained is used for understanding of tumor growth and proliferation. LCTFs have been used for studying retinal oxygen saturation in retina (Hirohara 2007). Moreover, combination of LCTFs with photostable, bright, narrow bandwidth, Quantum Dots-based fluorescent probes has been used for *in vivo* fluorescence imaging of small animals (Gao 2004) and in human tissues (Tholouli 2006). Last, LCTFs have been used in two-photon fluorescence microscopy (Lansford 2001), for discriminating mixtures of fluorescent proteins.

### 1.6.2 Acousto-optic Tunable Filter (AOTF)

An Acousto-optic Tunable Filter (AOTF) (Tran 2003, Stratis 2001, Bei 2004) consists of an acousto-optic crystal, a birefringent material whose optical properties can be altered by applying acoustic frequencies to the crystal. Crystals that are commonly used in the construction of AOTFs are Tellurium Dioxide ( $\text{TeO}_2$ ), Lithium Niobate ( $\text{LiNbO}_3$ ) and Calcium Molybdate ( $\text{CaMoO}_4$ ), among others. Operation of the AOTF relies on the interaction of light and acoustic waves traveling through a birefringent crystal. The density of the atoms making up the crystal is modulated by the acoustic or pressure wave. This results in modulation of the refractive index, which is directly affected by the structure of the atoms. Therefore, a volume grating is formed and moves with the speed of sound inside the crystal.

An AOTF works in so-called non-collinear configuration when the acoustic and optical waves propagate at quite different angles through the crystal. A piezoelectric transducer, bonded to the one side of the crystal, emits acoustic waves, usually at radio frequencies (50 to 200MHz). As these acoustic waves propagate through the crystal, they cause the crystal lattice to be alternately compressed and relaxed. The resultant density changes produce periodic refractive index variations via the elasto-optic effect. This periodic perturbation acts like a transmission diffraction grating, diffracting a narrow band of spectral frequencies at a time. The undiffracted wavelengths exit the crystal at the same angle as the incident light beam (zero-order beam). The diffracted wavelength is determined by the momentum matching condition for a transmission diffraction grating, described as

$$K_d = K_i \pm K_g$$

Where  $K_d$ ,  $K_i$  and  $K_g$  are the wave vectors of the diffracted beam, the incident beam and the acoustic wave respectively. The corresponding magnitudes of the vectors are  $K_d=2\pi n_d/\lambda_0$ ,  $K_i=2\pi n_i/\lambda_0$  and  $K_g=2\pi f/V$ , where  $f$  is the acoustic frequency,  $V$  is the acoustic speed in the crystal,  $\lambda_0$  is the centre wavelength and  $n_i$ ,  $n_d$  the indices of the refracted and diffracted beams respectively. Equation 3 describes a phonon-photon interaction in which photons of appropriate wavelength interact with phonons- the “packets of energy” transferred through the acoustic waves.

Figure 1.6.2.1 illustrates the principle of operation of a non-collinear AOTF. When unpolarized white light is incident on an AOTF, light of specific wavelength will be diffracted in two directions with orthogonal polarizations. One has Doppler up-shifted (+ diffracted beam) and the other has Doppler downshifted (-diffracted beam) optical frequency for the applied radio frequency. When the zero-order beam is used as output, the AOTF functions as a notch filter; if the  $\pm 1$ -order beam is used, the AOTF functions as a bandpass filter. The unwanted beams are rejected with a beam stop. Changing the frequency of the acoustic wave has the effect of changing the grating spacing, thus adjusting the wavelength of the diffracted beam. The wavelength of the diffracted beam is selectable at a bandwidth less than 2 nm by electronically varying the radio frequency. When multiple radio frequency signals are applied, multiple wavelengths are simultaneously diffracted in the same direction. This is an advantage over LCTFs, which generate only a single bandpass at a time. One should however keep in mind that AOTF devices require the incident light to be collimated.

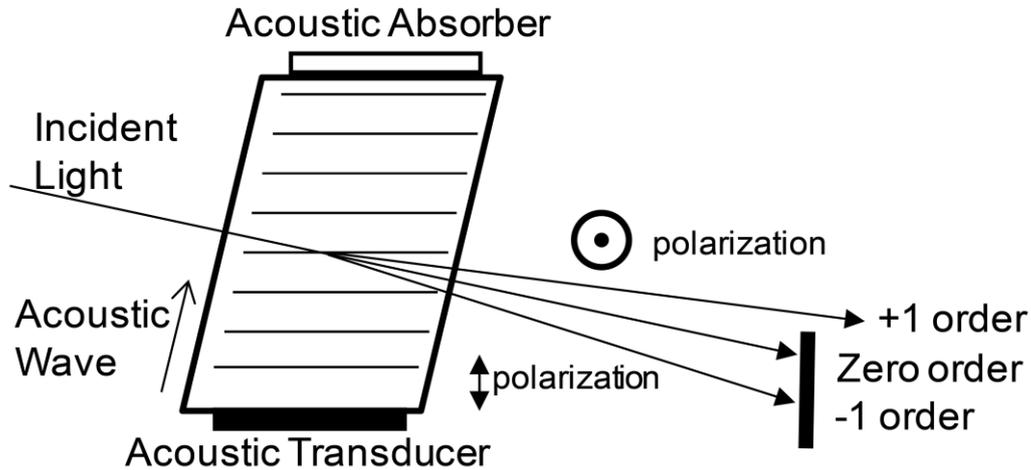


Figure 1.6.2.1 Schematic representation of an AOTF

The spectral operation range of an AOTF is quite extended and depends on the material used for the fabrication of the filter. For example, for  $\text{TeO}_2$  the operation range is 350-4500nm while for quartz crystal the range is extended to the UV range (120-6500nm). The bandwidth of the transmitted wavelength ranges from about 1nm in the UV range (using quartz crystal) to 15-25nm in the 1.5-2.5 $\mu\text{m}$  IR range (using the  $\text{TeO}_2$  crystal). Transmission efficiency is high, reaching the value of 98% when the incoming beam is polarized. Tuning speeds are often less than 50  $\mu\text{sec}$ , determined by the transit time of the acoustic wave inside the crystal.

The fact that the diffracted rays do not exit the acousto-optic crystal at a unique direction, as determined by eq.4.3, but over range of diffraction angles, causes the unwanted effects of image distortion and blurring. These affects restricted the maximum resolution to less than 1  $\mu\text{m}$  and this was for a period of time the main reason that the AOTFs were not widely used for imaging applications. The first attempt to correct this deficiency is dated back to 1996 when Wachman et al. (Wachman 1996) investigated the origins of AOTF image blurring. By employing image processing techniques the resolution increased at 0.35 $\mu\text{m}$ . Since then, several other methods have been applied for improving image quality involving optical correction (Wachman 1997, Suhre 2005) or digital image post-processing (Suhre 1998, Frances 2006).

AOTF-based spectral imagers have been used in several biomedical optical imaging applications (Rawja 2005, Treado 1992). Martin et al. (Martin 2006) demonstrated a dual-modality hyperspectral imager based on an AOTF, which was used for distinguishing normal from malignant mouse tissues. Wachman et al. (Wachman 1997) presented an AOTF-SI adapted to a microscope and recorded spectral images of oxyhemoglobin and deoxyhemoglobin in the brain of a living mouse. A dispersive prism was added in the experimental set-up to compensate for blurring effects. AOTFs have also been used in conjunction with fluorescence microscopes for multispectral image cytometry applications (Rajwa 2005), with confocal microscopes (Song 2008) and with endoscopes (Vo-Dinh 2004).

### 1.6.3 Optical Parametric Oscillator (OPO)

The last device that will be presented in this section is the optical parametric oscillator (OPO). Optically parametric oscillators are powerful solid state sources of broadly tunable coherent radiation covering the entire spectral range from the near UV to the mid IR and can operate down to the femtosecond time domain. As a result of recent advances in nonlinear optical materials research, these oscillators are now practical devices with broad potential applications in research and industry. Ever since the invention of the laser, there has always been a great deal of interest in the development of continuously tunable coherent light sources. Such sources would have broad applications in research and industry. The development of tunable oscillators has been difficult because conventional lasers tend to be discrete-wavelength devices involving stimulated emission between quantized energy levels in the laser media. Only when these quantized energy levels are tunable or there are neighboring energy levels that are sufficiently broadened to merge into each other to form a continuous band can a continuously tunable laser be built. Even then, the tuning range tends to be limited [LINKOPO1].

The optical parametric process is a nonlinear optical process in which a pump photon, propagating in a nonlinear optical crystal breaks down into two lower-energy photons with different frequencies. The crystal is placed inside an optical cavity (resonant Figure 1.6.3.1) and thus optical gain for parametric amplification is produced. So the difference between OPOs and lasers is that the optical gain of the output light is not based on stimulated emission. The main principle of the OPO operation is that the frequencies of the two photons produced (signal and idler) can be tuned in a wide wavelength range depending on the phase matching condition.

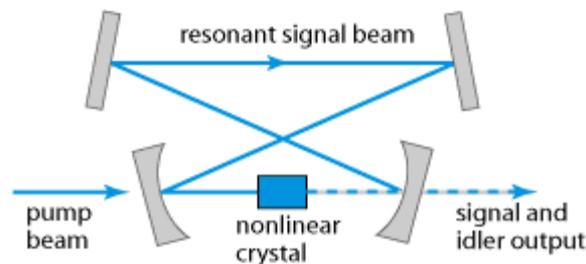


Figure 1.6.3.1 Typical OPO configuration

Phase matching is defined as a group of techniques for achieving efficient nonlinear interactions in a medium. Typically the phase matching in OPOs is ensured by the use of birefringent materials. The table below presents the tuning range of some OPOs according to the nonlinear optical crystal used.

Crystal	OPO tuning range (nm)
LBO <sup>1</sup>	415 – 2500 ( $\lambda_p = 355$ )
BBO <sup>2</sup>	415 – 2500 ( $\lambda_p = 355$ )
KTP <sup>3</sup>	577 – 4400 ( $\lambda_p = 532$ )

Table 1.6.3.1 Tuning range of various OPOs

---

<sup>1</sup> Lithium triborate

<sup>2</sup>  $\beta$ -barium borate

<sup>3</sup> Potassium titanyl phosphate

## Chapter 2

### Hardware Configuration

The goal of this thesis is to assemble and drive a hardware configuration that works as an excitation/emission microscopy device. Our main concern in the project was to construct a tunable light source using a linear filter which is driven by a stepper motor. This tunable light source is able to illuminate (excite) a microscope specimen with a wide range of bands (426nm – 796nm) and high output intensity. Also it provides a fast scanning of the whole range of the filter with a minimum of 1nm interval.

The next figure (Figure 2.1) shows a general overview of our excitation device. The components that are used in our system are depicted in Figure 2.1. Their description and analysis will be the subject of this chapter and the following paragraphs.

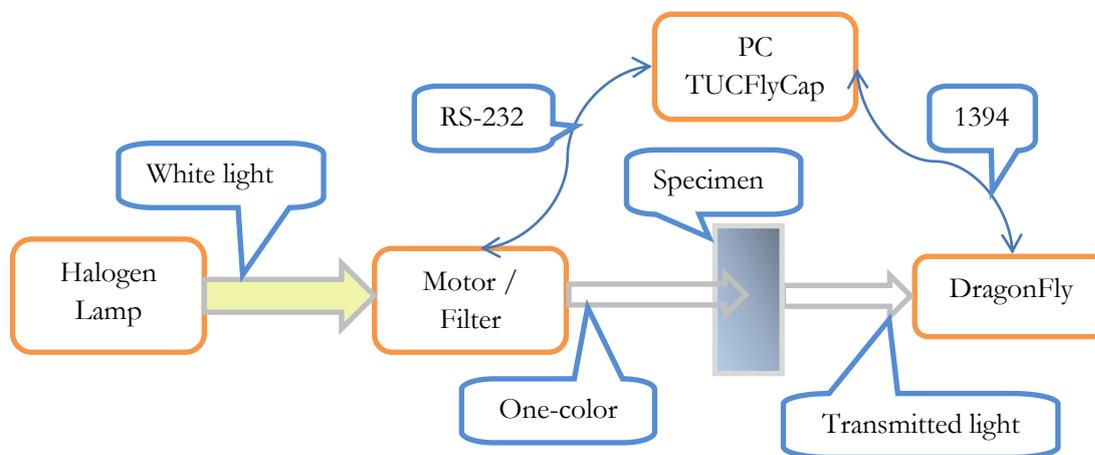


Figure 2.1 System Overview

As the “powering” component of the tunable light source we used a halogen lamp white light source (paragraph 2.1). The white light emitted by the halogen lamp is filtered by a linear filter (paragraph 2.2) which is mounted on a ramp and moved by a bipolar stepper motor (paragraph 2.3). The filtered light, which ideally is monochromatic or with narrow bandwidth, passes through a specimen of a microscope (paragraph 2.4). The transmitted light is captured and recorded as spectral data from a spectral imager/camera (paragraph 2.5). Both the filter movement and the camera settings are controlled via the same software using RS-232 communication for the filter and IEEE 1394 (FireWire) for the camera. The software is described and analyzed in Chapter 3.

## 2.1 White Light Source

In this paragraph we will present the output characteristics and performance of the white light source used as the illumination component of our system. We use the light source in order to produce a white light beam which is guided onto the filter through fiber optics. The next image (Figure 2.1.1) presents the specific light source we used in this project, NAVITAR 1-60563. This is a 220 volt fiber optic power supply with a 150 Watts EKE halogen lamp and a 0.720" fiber receptacle.



Figure 2.1.1 Navitar halogen light source 1-60563

- 1) **Fiber receptacle**, the part of light source marked is the socket for the bundle (one end) of the fiber optic from where the white light beam is inserted into the fiber.
- 2) **Intensity dimmer**, the light source incorporates a solid state dimmer to control the intensity of the output light.

As aforementioned the light source has a 150 W EKE halogen lamp which is depicted in Figure 2.1.2. A critical characteristic of the device we had to study was the emission spectrum of the lamp. Usually a halogen lamp (or a tungsten halogen lamp) will have an emission spectrum very similar to the one depicted in Figure 2.1.3. But many halogen lamps have special layers of materials that limit the emission spectrum. Measuring that spectrum would enable us to observe the emission bandwidth of the source and help us define how much the effective filter range will be. Making the assumption that the emitted light of our lamp is greatly reduced we conclude that the range of the filter used (effective range) will be equal to or smaller than the spectral emission range of the halogen lamp.



Figure 2.1.2 EKE 150W halogen lamp

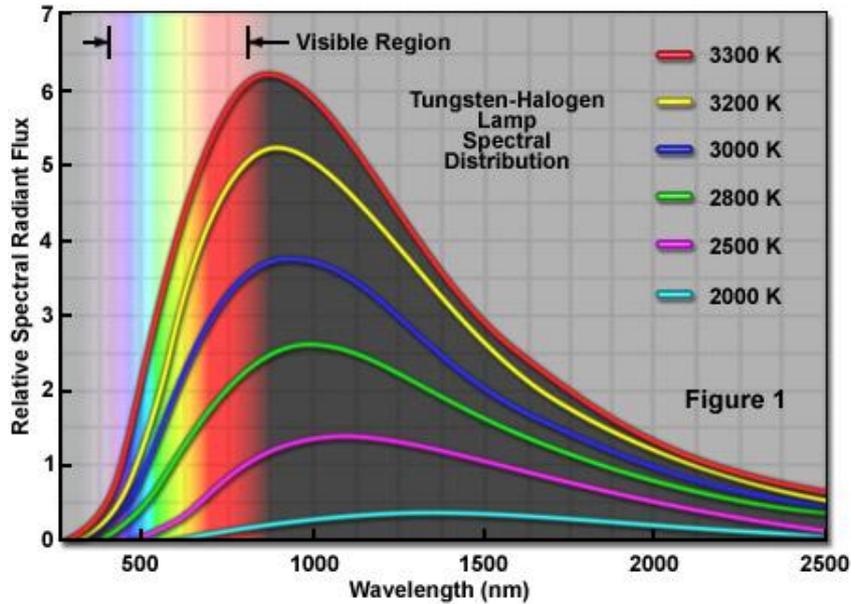


Figure 2.1.3 Usual Tungsten-Halogen lamp spectrum

As seen in the Figure 2.1.3 above a halogen lamp emits light whose spectrum extends into deep infrared wavelengths. To confirm that the bulb used in our light source produces the same spectrum we measured it using a spectrometer (Ocean Optics, USB4000) and the relative software (Spectra Suite). The measured spectrum is depicted in the next figure (Figure 2.1.4).

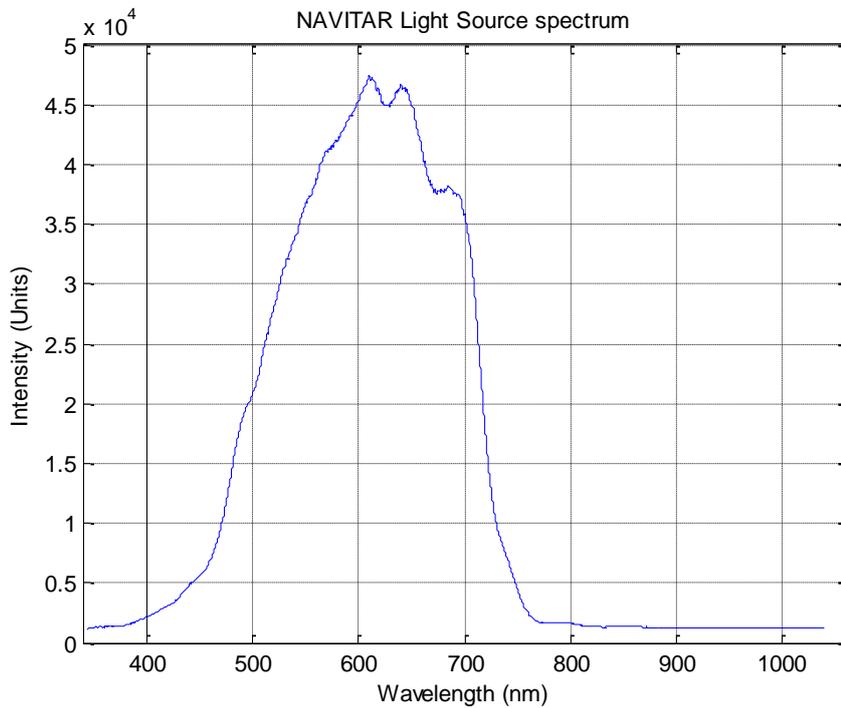


Figure 2.1.4 Measured spectrum of our halogen lamp

The measurement of the spectrum above was made with the output intensity tuned to the half of the scale provided by the dimmer.

It is clear by observing the light source spectrum that the output light hardly reaches the infrared band. The cut-off noticed near the beginning of the infrared band is due to the optical coating inside

the “cone” of the lamp. An optical coating is one or more thin layers of material deposited on an optical component such as a lens or mirror (in our case on the mirror/reflector of the lamp), which alters the way in which the optic reflects and transmits light. One type of optical coating is an antireflection coating, which reduces unwanted reflections from surfaces, and is commonly used on spectacle and photographic lenses. Another type is the high-reflector coating which can be used to produce mirrors which reflect greater than 99.99% of the light which falls on them. More complex optical coatings exhibit high reflection over some range of wavelengths, and anti-reflection over another range, allowing the production of dichroic thin-film optical filters.

A dichroic reflector is used on our halogen lamp. The inside of the lamp “cone” is covered with dichroic reflector which reflects the visible light forward while allowing the infrared light (radiated heat) to pass out of the rear of the lamp. This helps keep the beam of the light “cooler”, meaning with reduced thermal noise which is caused by the infrared light.

So the infrared emission of the halogen lamp is cut-off by the dichroic reflector coating, but since the linear filter has the ability to filter infrared radiation (740nm – 1000nm) it is in our interest to use a light source that emits at infrared bands. That way we could excite the microscope specimens with infrared radiation and we could collect more useful data about those specimens.

One notion we examined in order to have infrared emission, was to try to extend the spectrum of the lamp more into the infrared zone by manually altering some areas of the lamp’s coating. The resulted coating of the lamp should be more reflective in the infrared band in order to increase the infrared radiation of the output light. Since reflectivity was an objective of the coating we wanted to alter, we chose to deposit gold reflective attachments on the “cone” of the lamp. As shown in Figure 2.1.5 gold is one of the most reflective metals especially in the infrared region.

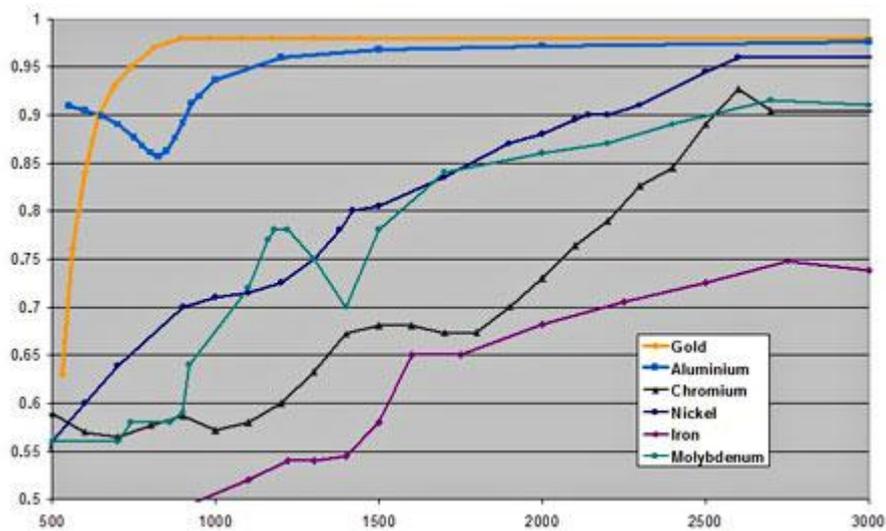


Figure 2.1.5 Reflectivity of metals (Normalized reflectivity over wavelength (nm))

Golden reflectors are used in the development of infrared heaters and in space exploration applications for temperature due to gold’s higher and more consistent reflective properties. The coating we added was gold wrapping paper, that way we would reduce the efficiency of the dichroic reflector and increase the amount of the infrared in the output light. The gold coating was attached on the reflector as depicted in Figure 2.1.6.



Figure 2.1.6 EKE lamp, gold coating

Unfortunately this effort didn't produce the expecting results. As seen in Figure 2.1.7 the spectrum of the gold-coated lamp is reduced in terms of intensity and is not expanded at all in the infrared bands compared with the original one.

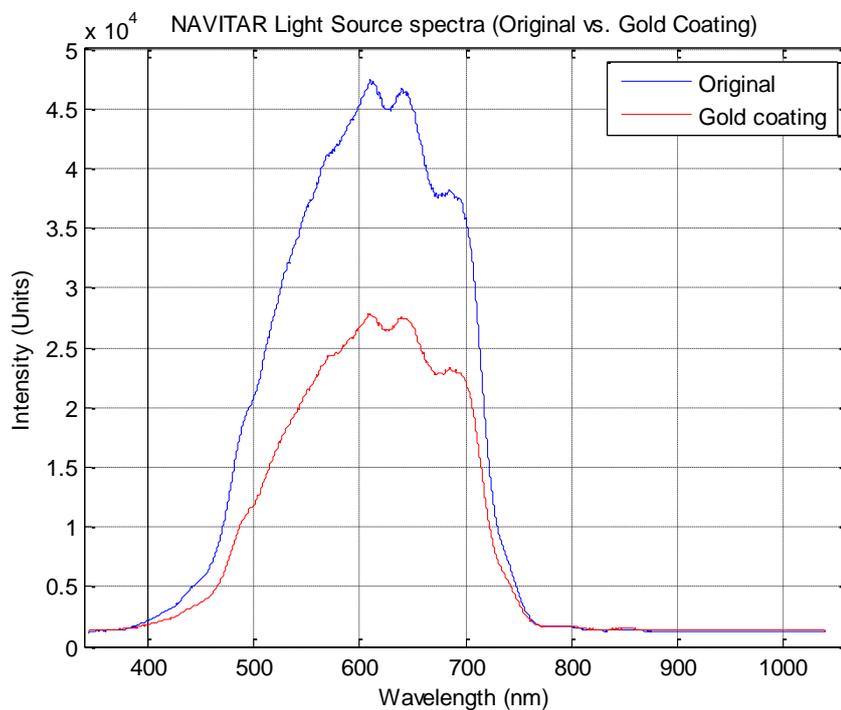


Figure 2.1.7 Light Source Spectrum, gold coating

The failure of the above coating effort is probably due to the lack of “proper” gold material attached on the dichroic reflector. The use of wrapping paper is not the best option for the implementation of the coating idea. But since we had no other 150 W halogen lamp with infrared emission, we decided to continue our configuration using this Navitar light source as it is and to limit the use of the linear filter to a smaller spectral range (426nm – 796nm).

The last part of the illumination device of our tunable excitation system that is presented in this paragraph is the optical connection between the light source and the linear filter. This connection is achieved with the use of fiber optics as line light guides. In Figure 2.1.8 the two (one input, one output) fiber optics are depicted mounted on the system.

The light enters the fiber from the fiber's bundle as it exits the receptacle of the light source. Then it travels to the other end of the fiber where it is reformed to a linear shape (slit). The slit exits the fiber and it is focused on the filter's surface by special cylindrical lens configuration (Figure 2.1.8 – 1). The

filtered slit of light is received by the opposite aligned cylindrical lens configuration (Figure 2.1.8 – 2) that converts the slit to a focused light beam that enters the output fiber light line guide.

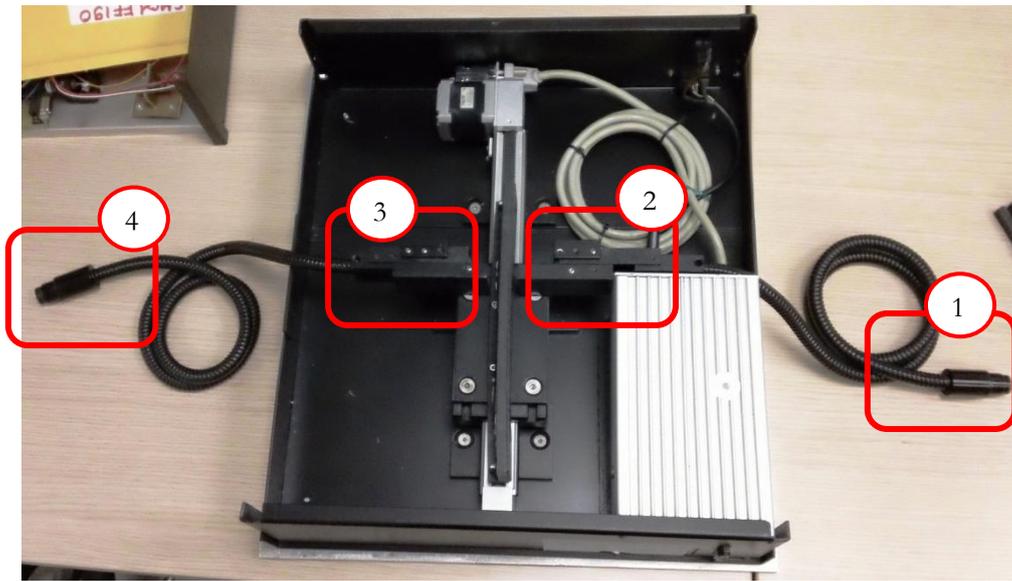


Figure 2.1.8 Optical connection of the light source with the filter and the system's output

- 1) **Input fiber bundle**, the output light of the source enters the fiber and travels towards the filter.
- 2) **Input cylindrical lens configuration**, the light exits the line guide and it's focused on the filter as a 2mm slit. From beam to slit.
- 3) **Output cylindrical lens configuration**, the filtered light enters the line guide focused by the mirrors and transforms to beam. From slit to beam.
- 4) **Output fiber bundle**, the filtered light exits the fiber and illuminates the microscope specimen.

Next we analyze the characteristics of the filter used to change the output light wavelength.

## 2.2 Linear Variable Band Filter

The tunability of our light source depends on two components; the filter used to produce (ideally) monochromatic light and the motor used to move the filter, thus changes the wavelength of that monochromatic output. In this paragraph we will present the main characteristics of the linear filter we use in this project.

The filter selected for this thesis is a Linear Variable Filter (LVF); Schott, VERIL BL200. The filter is shown in Figure 2.2.1 mounted on the ramp moved by the stepper motor.



Figure 2.2.1 Linear variable bandpass filter, Schott – VERIL BL200

A LVF is a bandpass filter where the coating has been intentionally wedged in one direction. This wedge causes the center wavelength ( $\lambda_m$ ) of the filter to shift linearly across the length of the filter. Adjusting the filter orientation allows a specific wavelength to be selected. The VERIL filters possess the same curve characteristics as the corresponding homogeneous filters. Additional blocking is brought about in some cases by graduated colored glasses. When linear variable filters with a pre-fitted slit are used, increasing the slit width widens the passband curve and reduces the maximum transmittance. Slit widths up to 3 mm in the case of BL 200 have practically no effect on spectral performance.

A critical parameter of the LVF is the transmission of the filter, meaning the response of the LVF according to the wavelength of the excitation light slit. This parameter provides the designer with useful information about the behavior of the filter. It is important to be aware in which bands the filter transmits more intense light and is also important to know the full width half maximum (FWHM) parameter.

Full width at half maximum (FWHM) is an expression of the extent of a function, given by the difference between the two extreme values of the independent variable at which the dependent variable is equal to half of its maximum value. FWHM is applied

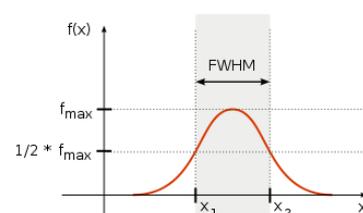


Figure 2.2.2 FWHM example

to such phenomena as the duration of pulse waveforms and the spectral width of sources used for optical communications and the resolution of spectrometers.

The Figure 2.2.3 presents the bandpass transmission of the filter measured with the center wavelength from one end of the filter (400nm) to the other (1000nm) with 20nm interval. This figure is initially presented in the filter’s datasheet without any commentary from the constructor. But using this figure and observing Table 2.2.1, which presents some key values of the filter we will have a better view about the filter’s performance.

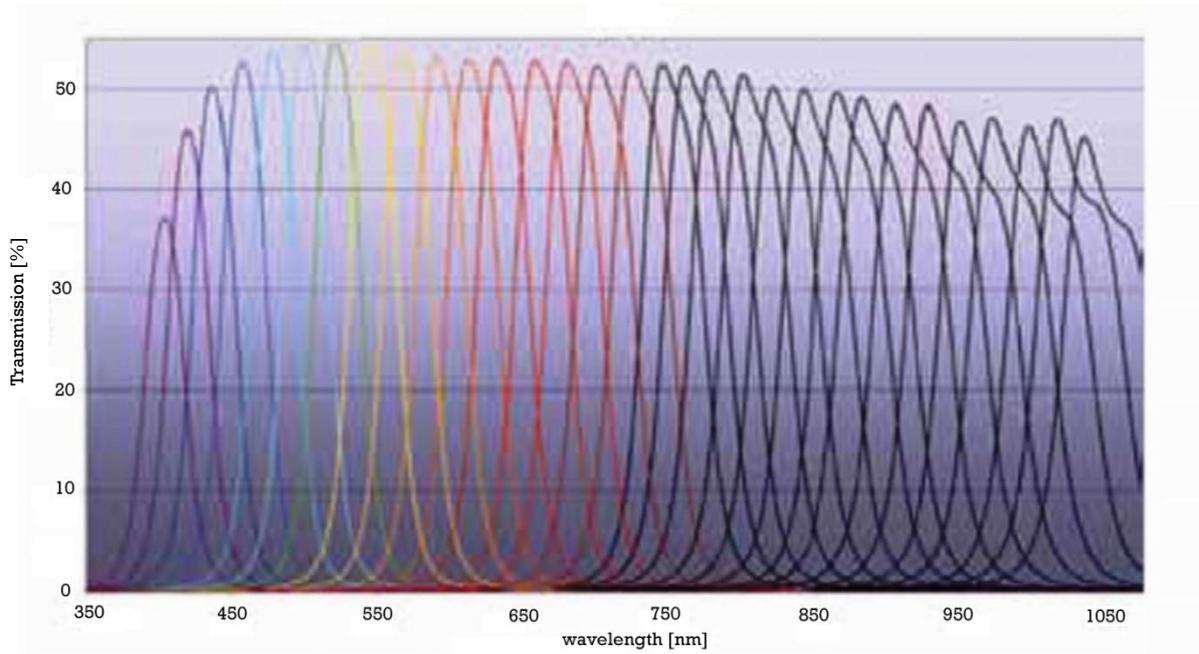


Figure 2.2.2 Transmission spectra of the LVF, Schott – VERIL BL200

Parameter	Unit	Value
Spectral range of $\lambda_m$	[nm]	400 – 1000
Reciprocal linear dispersion FWHM	[nm/mm]	3.6 – 4.4
- $\lambda_m = 500\text{nm}$	[nm]	25 – 45
- $\lambda_m = 700\text{nm}$	[nm]	35 – 50
- $\lambda_m = 900\text{nm}$	[nm]	40 – 65
Maximum transmittance $\tau_{max}$		
- $\lambda_m = 500\text{nm}$	--	$\geq 0.4$
- $\lambda_m = 700\text{nm}$	--	$\geq 0.4$
- $\lambda_m = 900\text{nm}$	--	$\geq 0.3$
Blocking range	[nm]	unlimited
Operating temperature	[°C]	70 °C, several hours 100 °C, short periods

Table 2.2.1 Main characteristics of LVF, Schott – VERIL BL200

Analyzing the above information that the transmission figure and the characteristics table provide, we can come to various conclusions. Firstly, we can see that the LVF provides a big range of possible wavelengths as output of our system. Also that range is consistent in terms of wavelength dispersion across the length of filter. Ideally a LVF with 200mm length and 600nm spectral range should have a

3nm/mm “density” of center wavelengths across its length. As presented above the dispersion of this parameter is close to the ideal value and does not have a variance that can affect its linearity. Secondly, if we take a closer look at the FWHM values we can see that the best case of a filtered spectrum’s FWHM would be the 4.16% of the total spectral range. The worst case of a FWHM would be the 10.83% of the total spectral range. Those attributes do not rank our LVF as an ideal component for our system since the desired output is monochromatic light. Lastly, this LVF is one of the few that can withstand a high operation temperature under a long operation period. The high temperature will be caused by the light source (150 Watts) and if the source is emitting in the infrared region the heating will be a more important system parameter.

## 2.3 Stepper Motor and the Driving Mechanism

In this paragraph we describe a critical component and its function. The mechanism which moves the LVF is maybe the most important to understand in order to drive the whole mechanism efficiently. An overview of this component is presented in the next figure (Figure 2.3.1).

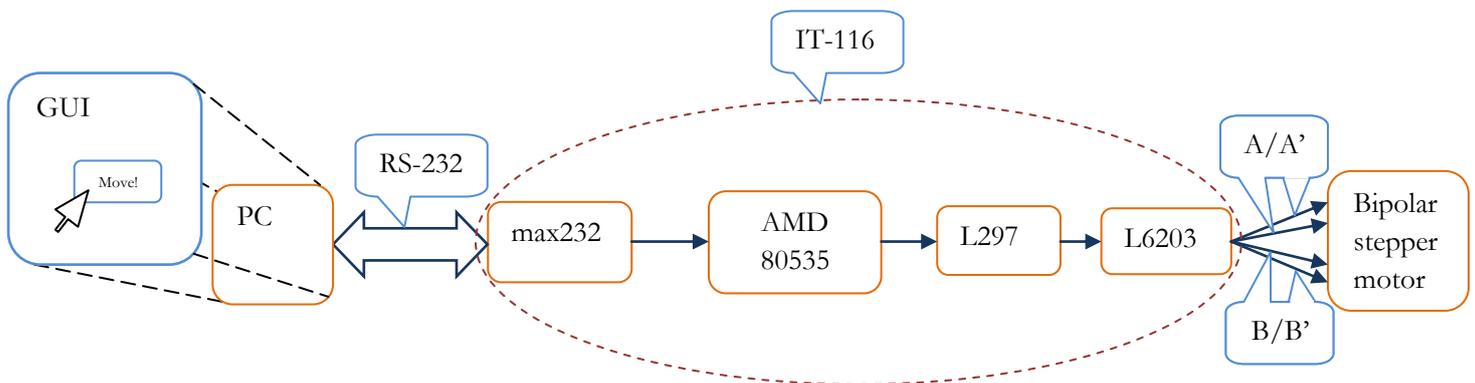


Figure 2.3.1 Filter moving mechanism overview

The filter movement is ignited from a user instruction given to the PC, using a GUI that is analyzed in Chapter 3. Then the instruction is translated and sent to the controlling subcomponents which are depicted above and described in the next paragraphs. These subcomponents drive a bipolar stepper motor which executes the given instruction. The structure and the function of the stepper motor used are analyzed further below.

### 2.3.1 Connecting with the serial port

As aforementioned the user instruction is passed to the stepper motor firstly through a serial port connection using the RS-232 standard. More specifically the user instruction is given as input to a specific stepper motor controller that is used; ISEL – IT-116. A few key components of this controller are depicted in the overview above and will be analyzed in the next sections.

The first step to communicate with the controller is to configure the serial port communication with the correct properties, which are specified by the controller’s manual. In order to establish serial communication we need to use a customized serial cable which follows specific pin mapping similar to

the Null modem method. Null modem is a communication method to connect two DTEs [Data Terminal Equipment] (computer, terminal, printer etc.) directly using an RS-232 serial cable. The name stems from the historical use of the RS-232 cable to connect two teleprinter devices to modems in order to communicate with one another; null modem communication was possible by instead using RS-232 to connect the teleprinters directly to one another. A null modem cable is an RS-232 serial cable whose transmit and receive lines are cross-linked. In some cables there are also handshake lines cross-linked. A common pin mapping is shown in Figure 2.3.1.1.

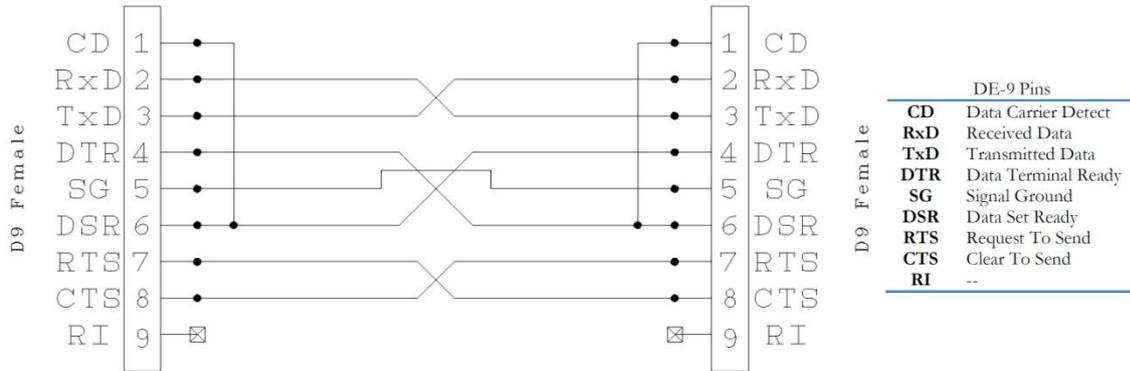


Figure 2.3.1.1 Standard DE-9 cable pin mapping

The IT-116 stepper controller as aforementioned has a prerequisite of a specific cable connection which is presented in the figure below (Figure 2.3.1.2).

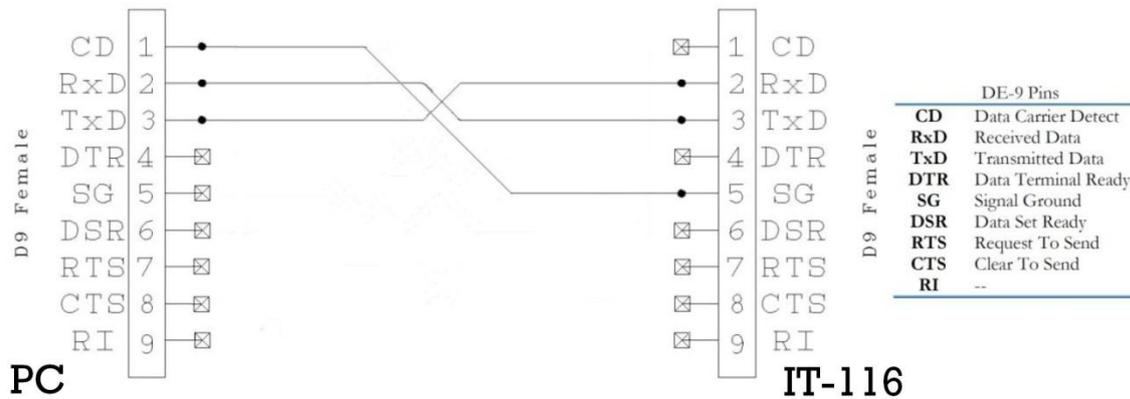


Figure 2.3.1.2 Customized DE-9 cable pin mapping for ISEL, IT-116

As it is shown the customized cable does not have many connections, there is no handshaking implemented and the signal ground is not common for the two terminals. The connections that remain the same are the TxD and RxD signals which are cross-linked. These customizations are made because there is no synchronized connection between the PC and the controller. The data have more of an explosive nature. Meaning that there are no data transferred either from the PC or from the controller for long periods of time. The signal ground (SG) of the IT-116 is enabled only when data are transferred from the PC. This is because the controller is directly connected to the plug (220 Volts) and has its own electrical transformer (220 V AC to 40 V DC) to power the rest of the controller's circuits. So the ground of this high voltage transducer should not be connected with the ground of another lower voltage system, such as the serial port of the PC.

After establishing the serial connection with the correct cable wiring we have to study the function of some important components inside the IT-116 controller.

### 2.3.2 Stepper controller Breakdown

In this section we will analyze some key areas of the IT-116 operation. As shown in Figure 2.3.1 the stepper controller we use has four significant components; a dual EIA-232 driver/receiver (MAX232), an embedded controller (AMD-80535), a stepper motor controller IC (L297) and a DMOS full bridge driver (L6203).

The IC MAX232 (Texas Instruments) is used to translate the RS-232 signals. The standard specifies a maximum open-circuit voltage of 25 volts: signal levels of  $\pm 5$  V,  $\pm 10$  V,  $\pm 12$  V, and  $\pm 15$  V are all commonly seen depending on the power supplies available within a device. RS-232 drivers and receivers must be able to withstand indefinite short circuit to ground or to any voltage level up to  $\pm 25$  volts. So the MAX232 does just that, converts the signals' voltage levels from RS-232 to 5 Volts TTL/CMOS levels and vice versa.

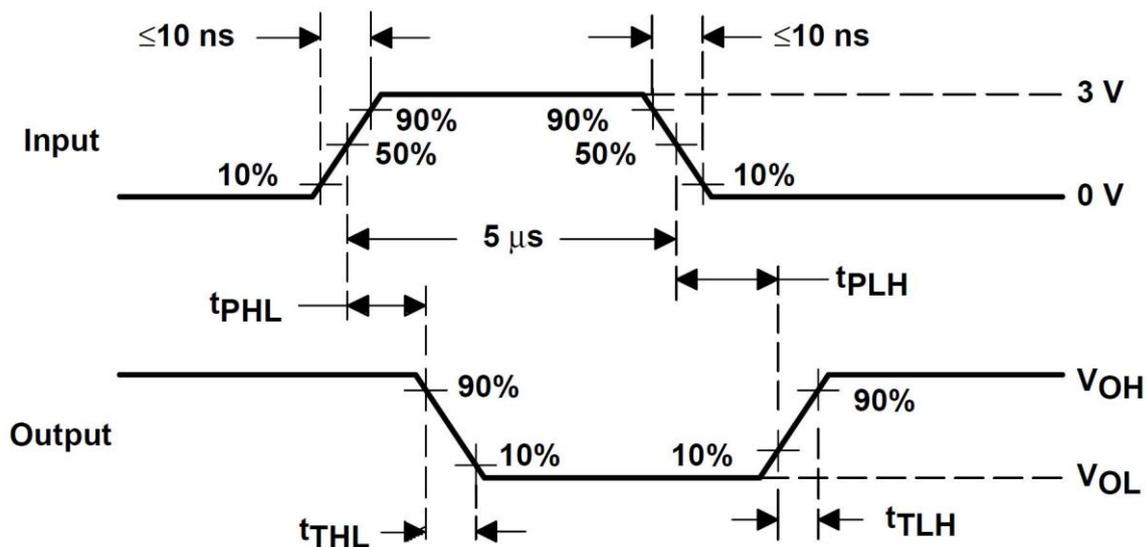


Figure 2.3.2.1 MAX232 operation, from TTL to RS-232

- In the figure above:
- $V_{OH}$  == High-level output voltage
  - $V_{OL}$  == Low-level output voltage
  - $t_{PHL}$  == Receiver propagation delay time, high-to-low level output
  - $t_{PLH}$  == Receiver propagation delay time, low-to-high level output
  - $t_{THL}$  == Receiver transition delay time, high-to-low level
  - $t_{TLH}$  == Receiver transition delay time, low-to-high level

The MAX232 is used by the microcontroller (AMD-80535) as a translator to send and receive data from/to the serial port of the PC.

AMD-80535 is an 8-bit, stand-alone, high-performance, single-chip microcontroller based on the 8051 architecture. The 805 35 is ideal for motor control applications ranging from automotive engines

to vending machines. 8051 (or Intel MCS-51) architecture is a Harvard architecture<sup>1</sup>, microcontroller series developed by Intel in the 1980 for use in embedded systems. Here below we cite some specifications/characteristics of AMD-80535:

- 256 x 8 RAM
- Six 8-bit ports; 48 I/O lines
- Full-Duplex Serial Port
- Twelve interrupt sources; four priority levels
- 8-bit A/D converter
- Most instructions execute in 1  $\mu$ s
- 64K bytes Program memory space
- 64K bytes Data memory space

The block diagram, the connection diagram and the logic diagram of the microcontroller are available in the processor's datasheet [[LINK](#)].

An interesting observation is that the AMD-80535 is used when the stepper controller is set in a stand-alone mode. IT-116 is equipped with such microcontroller so that it can drive a stepper motor by executing a program loaded and not by executing instructions given from an external CPU. Since in our system the stepper controller is constantly connected to a PC, we realize that the microcontroller is used only to set and broadcast the controller's clock to the rest of the driving circuit's components. Also it is used to translate the instruction transmitted through RS-232 and set the corresponding input values of the next key component (L297).

This stepper motor driver IC; ST Microelectronics – L297, is an IC that generates four phase drive signals for two phase bipolar and four phase unipolar step motors in (micro-)computer controlled applications. The motor can be driven in half step and normal drive modes and on-chip PWM chopper circuits permit switch-mode control of the current in the windings. The L297 IC is maybe the most common IC used for stepper motor driving applications and systems. Also many times this IC is encountered driving a monolithic bridge drive such as the L298N or L293E, or with discrete transistors and darlington<sup>2</sup>. The Figure 2.3.2.2 presents a common driving circuit of a bipolar stepper motor.

---

<sup>1</sup> The Harvard architecture is computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. Today, most processors implement such separate signal pathways for performance reasons but actually implement Modified Harvard architecture, so they can support tasks such as loading a program from disk storage as data and then executing it.

<sup>2</sup> The Darlington transistor (often called a Darlington pair) is a compound structure consisting of two bipolar transistors (either integrated or separated devices) connected in such a way that the current amplified by the first transistor is amplified further by the second one. This configuration gives a much higher current gain than each transistor taken separately and, in the case of integrated devices, can take less space than two individual transistors because they can use a shared collector.

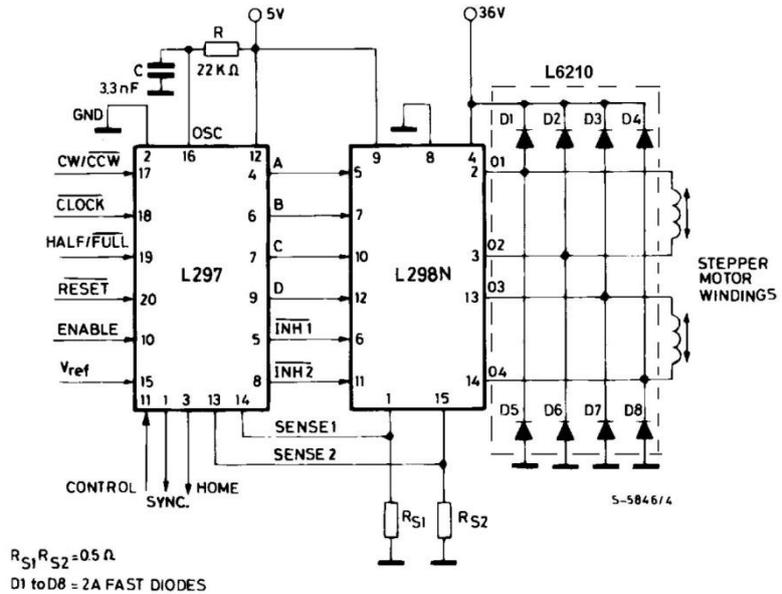


Figure 2.3.2.2 Common driving circuit of a bipolar stepper motor, L297 and L298

The circuit presented above is very close to what the IT-116 contains. Studying the characteristics of the L298 IC which is used as a bridge in the above circuit, we realized that this IC does not provide enough output current (provides 1A/output) to drive our bipolar stepper motor. This is why IT-116 does not use this IC as bridge but instead uses ST Microelectronics - L6203 DMOS full bridge driver. L6203 provides a total of 4A current output, meaning 2A/output because the IC has two outputs. Since a bipolar stepper motor has four inputs (two for each pole) IT-116 uses two ICs as depicted below. These two ICs are pinned on a thick metallic plaque for heat abduction.

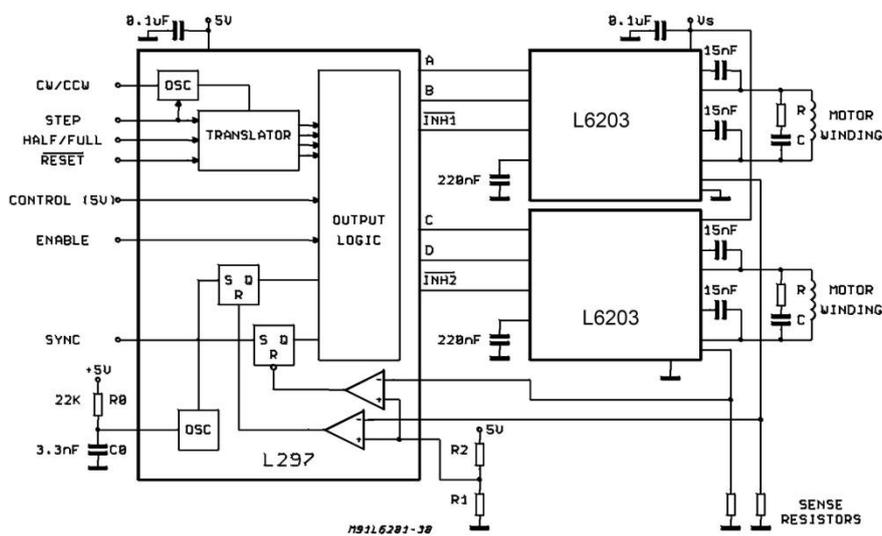


Figure 2.3.2.3 IT-116 driving circuit of a bipolar stepper motor, L297 and L6203

### 2.3.3 Bipolar Stepper Motor

The bipolar stepper motor is the driving/moving force of our configuration. Figure 2.3.3.1 depicts the bipolar stepper motor we used; NANOTECH – ST4118L1804. In this section we will analyze its operation and performance. The range, accuracy and responsivity of the stepper motor are important factors for the characterization of the overall performance of the tunable light source configuration we built.

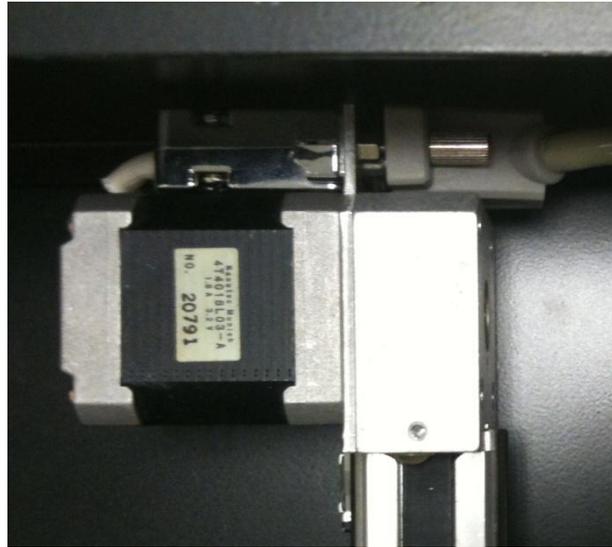


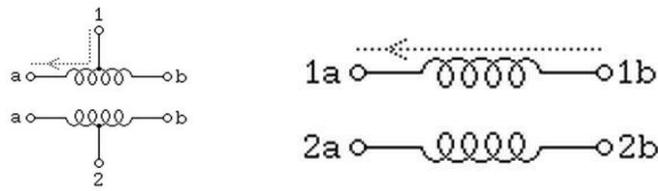
Figure 2.3.3.1 Bipolar stepper motor NANOTECH – ST4118L1804 mounted on our configuration

In order to understand the motor's performance we need first to mention some key principles of stepper motors' operation.

Stepper motors provide a means for precise positioning and speed control without the use of feedback sensors. The basic operation of a stepper motor allows the shaft to move a precise number of degrees each time a pulse of electricity is sent to the motor. Since the shaft of the motor moves only the number of degrees that it was designed for when each pulse is delivered, you can control the pulses that are sent and control the positioning and speed. The rotor of the motor produces torque from the interaction between the magnetic field in the stator and rotor. The strength of the magnetic fields is proportional to the amount of current sent to the stator and the number of turns in the windings.

[\[LINK\]](#)

Stepper motors are either bipolar, requiring two power sources or a switchable polarity power source, or unipolar, requiring only one power source. They are powered by DC current sources and require digital circuitry to produce the coil energizing sequences for rotation of the motor. Unipolar stepping motors with 5 or 6 wires are usually wired as shown in the schematic in Figure 2.3.3.2a, with a center tap on each of two windings. In use, the center taps of the windings are typically wired to the positive supply, and the two ends of each winding are alternately grounded to reverse the direction of the field provided by that winding.



a) Unipolar stepper motor    b) Bipolar stepper motor

Figure 2.3.3.2 Stepper motor windings

Bipolar permanent magnet motors (Figure 2.3.3.2b) are constructed with exactly the same mechanism as is used on unipolar motors, but the two windings are wired more simply, with no center taps. Thus, the motor itself is simpler but the drive circuitry needed to reverse the polarity of each pair of motor poles is more complex. The bipolar stepper motor we use is more specifically wired as follows.

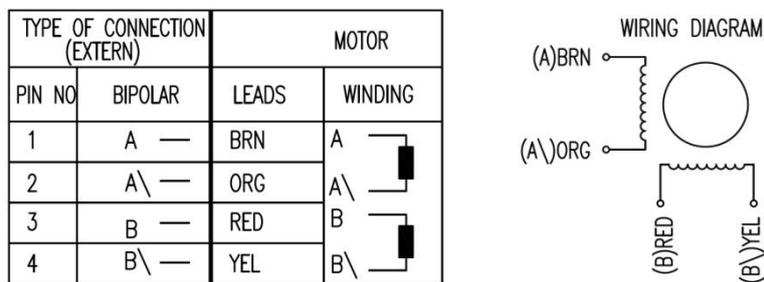


Figure 2.3.3.3 ST4118L1804 wiring

So the motor is connected with the stepper controller with four cables (two for coil A and two for coil B). The cables are passed through a DE-9 interface shown in top right of Figure 2.3.3.1. The current of each cable are controlled and provided by the L297 and L6203 circuit.

The most important issue that is imposed by the use of a stepper motor is the delays that causes. When we move the filter to a desired wavelength this movement is executed through the stepper motor, which inserts a delay until the movement is completed. In order to characterize a part of our system performance we thoroughly studied the delays that can be inserted by the stepper motor. Firstly we constructed a little piece of software that sends instructions to the motor and registers its duration (in ms) by reading the completion signal that is fed back from the stepper controller. Then we run that software for a big number of repetitions (574) with randomly chosen instructions (random positions). The results of these measurements are presented in Figure 2.3.3.4.

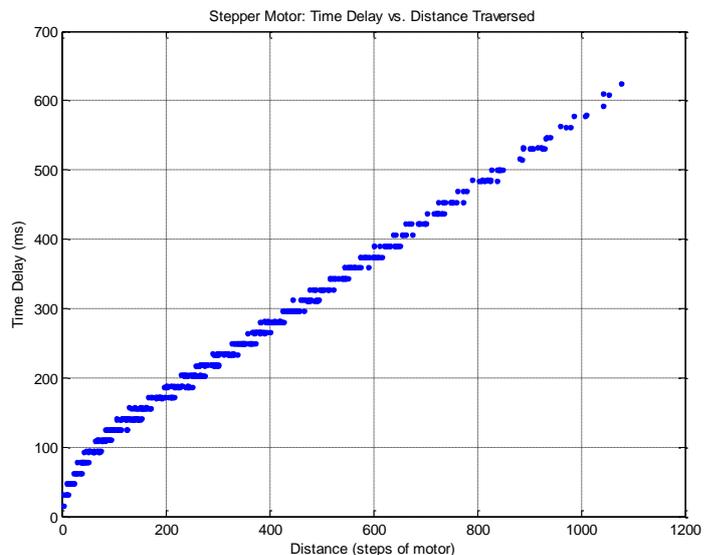


Figure 2.3.3.4 Bipolar stepper motor's delay

As it is shown in the figure above the delay of the stepper motor to reach the desired position is almost linear in proportion to the distance it has to traverse. This observation is something expected but the interesting issue that is presented above is that the motor's delay is quantized into specific levels, which are overlapped. Meaning that a movement to a certain position may endure either level(n) milliseconds or level(n+1) milliseconds. This quantization is caused by the nature of the stepper motor, since the motor is driven by pulses and each step is result of a current pulse it is logical for the delay increment to have a stepping nature.

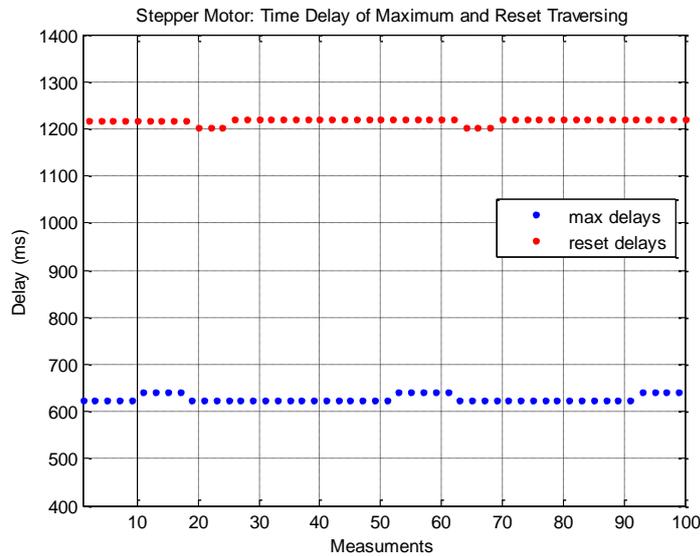


Figure 2.3.3.5 Stepper motor's delay of maximum and reset traversing

Another aspect of the motor's delay we measured was the maximum possible delay (maximum distance traversed) and the delay that the motor made coming to its Reset position from the maximum end. As it is noticed the maximum position is reached with a delay around 630 ms, but the return to the reset position lasts approximately double amount of time (1200 ms). This is because the Reset position is reached when the filter hits a metallic and a stop signal is produced. Then the motor steps one step away from the stop sensor so that they do not have contact. Since the Reset procedure is more complex (because of the stop sensor) than the movement to a certain position it is expected for this instruction to delay the longer.

Although it is useful to know which are the limits and the extreme values of the delay that is inserted to our system by the stepper motor it is more important to know how much the motor's delay is affecting the system when a scan of an area of the filter is instructed. The scanning procedures (as explained in Chapter 3) are the most common operations for our system and also the most important. When the motor is moving the filter accordingly to scan certain bands (wavelengths) of the filter it is unlikely to have long traverses as the ones in Figure 2.3.3.5. It is more likely that the step of the scanning procedure would be mainly a short distance that can be quickly traversed (10nm – 20nm). The affect of these delays on our system's performance are discussed in Chapter 5 - System Evaluation.

## 2.4 Microscopy Configuration

The filtered light coming out of our configuration that we have described so far is given as input to the microscope of our system. We use an upright microscope for transmission and fluorescence microscopy; OLYMPUS – BX51, which is depicted in Figure 2.4.1



Figure 2.4.1 OLYMPUS – BX51 for transmission/fluorescence microscopy

The above depicted microscope has the ability of operating in two modes. The first is the common transmission microscopy where the light (generated from a halogen lamp located at the lower-backside of the microscope) is transmitted through the specimen and then it is depicted at PC monitor using a digital camera and/or seen by from the ocular (eyepiece) lens. The second mode is the fluorescence (or epifluorescence) mode, where the light is inserted using a “special” lamp (arc lamp; mercury or xenon lamp) by the lamphouse located at upper-backside of the microscope. Figure 2.4.2 presents an explanatory scheme of epifluorescence microscopy.

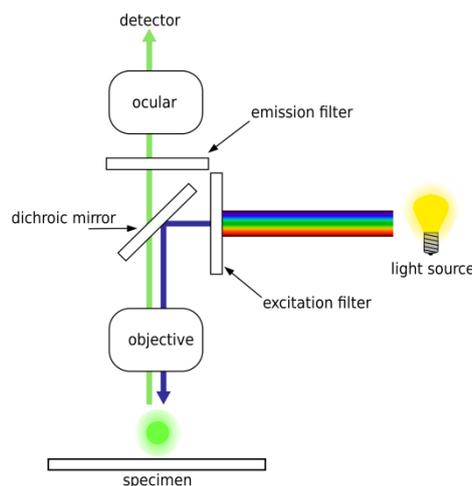


Figure 2.4.2 Schematic of a fluorescence microscope

Since the goal of this project is to build an excitation/emission microscopy configuration and more specifically to assemble and drive the tunable excitation source we did not use the fluorescence microscopy mode of the microscope. We use the transmission mode in order to achieve higher reflectance for better spectral data. So the output fiber bundle of the tunable light source is inserted in the receptacle of the lower-backside lamphouse, as depicted in Figure 2.4.3.



Figure 2.4.3 Connection of the fiber bundle to the lower lamphouse

### Objective lenses

In the process of testing our configuration and also when acquiring spectral data that are used in Chapter 4 where overall results of this configuration are presented, we used mainly two objective lenses; OLYMPUS – UIS2, UPLFLN 40x and OLYMPUS – UIS2, PLN 10x. These two lenses are shown in Figure 2.3.4.4 below.

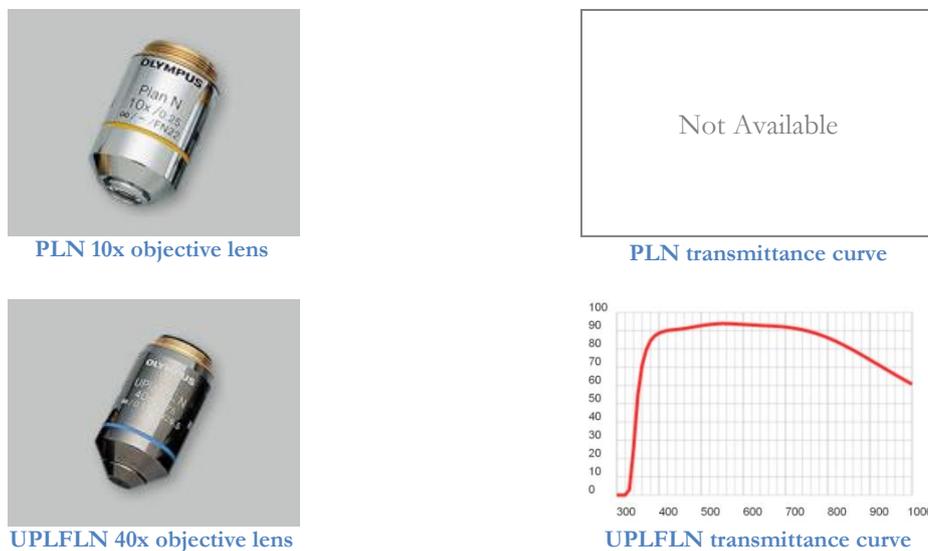


Figure 2.4.4 Objective lenses used in our system

Mostly when acquiring spectral cubes with our configuration we used the PLN 10x objective because it has greater transmittance, thus the image registered by the camera has greater reflectance and the spectral data are more clear than using UPLFLN 40x. In the figure above there is no transmittance curve presented for PLN 10x objective and this is because OLYMPUS does not provide us with such a curve for the specific lens, only with specifications and dimension information.

## 2.5 Spectral Camera

The last, but not least, part of our hardware configuration is the spectral imager. The spectral camera is what eventually combines spectroscopy and imaging into spectral imaging. The selection of this device is obviously very crucial, since it defines in a significant manner, the quality of the spectral data acquired. The maximum resolution of the camera, the frame rate (fps) and the shutter and gain capabilities affect the flexibility and the usability of a spectral imaging system. Also the characteristics of the sensor used in the camera do determine the quality of the spectral imaging system.

In this project we used POINT GREY RESEARCH (PGR) – DragonFly2-13S2C-CS. This device is depicted here below (Figure 2.5.1) and is a full featured IEEE 1394a (FireWire) digital spectral camera. Its CCD sensors can acquire both Black & White (BW/gray-scale) and color images.



Figure 2.5.1 PGR – DragonFly2-13S2C-CS

The device depicted on the right is a bare-bone (no cover) version of the camera. Actually in this project we used a FORTH PHOTONICS – MuSIS MS, which is a multi-spectral imaging device that incorporates the PGR camera. This device has a rotating wheel of several (8) filters in front of the camera’s CCD in order to acquire spectral data of different bands. Because we have our own filtering configuration before the light is transmitted through the microscope specimen, we use the MuSIS MS without filters, only as a DragonFly 2 camera.

The next table presents some key specifications of the spectral camera.

Parameter	Value
Image Sensor Type	Sony 1/3” progressive scan CCD
Image Sensor Model	ICX445AQ
Sensor Pixel Size	3.75 $\mu$ m square pixels
Maximum Resolution	1296x964
Maximum Frame Rate (fps)	1296x964 at 20 fps
Gain	Automatic/Manual/One-Push modes, 0dB to 24dB
Shutter	Automatic/Manual/One-Push/Extended modes 0.01ms to 66.63ms at 15 FPS, greater than 5s in extended mode

Table 2.5.1 PGR – DragonFly2-13S2C-CS specifications

From Table 2.5.1 we can see that the resolution provided and the overall performance of the camera are adequate for our system since we can operate in high resolution with relatively fast (real-time for biology/microscopy imaging) frame rate. The last issue we have to present is the CCD sensor that the DragonFly 2 camera uses.

To process data from digital color cameras, it is often necessary to know the spectral response properties of the camera's sensors. For example, the intrinsic color quality of an image depends on the spectral characteristics of the sensors [\[LINK-HORN\]](#) used. Thus an important component of characterizing a digital color camera is measuring its sensor spectral response functions. [\[LINK-HP\]](#) Figure 2.5.2 presents the CCD's (ICX445AQ) spectral sensitivity characteristic.

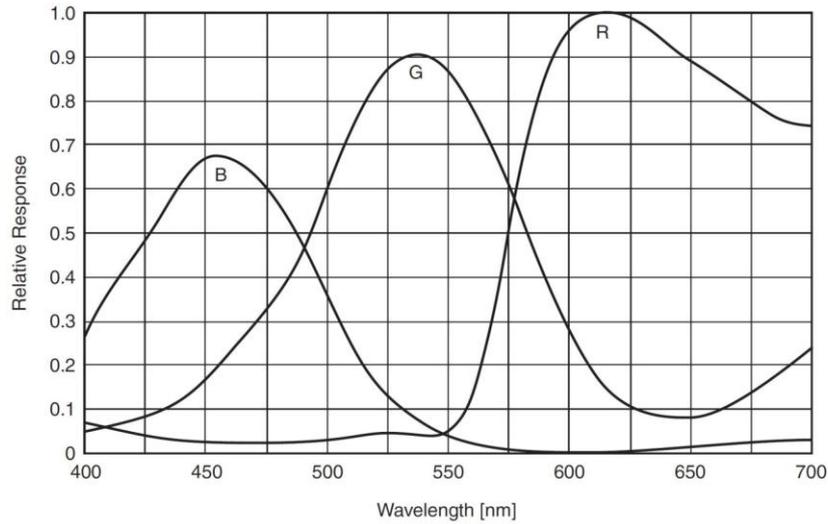


Figure 2.5.2 Spectral sensitivity characteristics of ICX445AQ

The sensitivity presented above is independent of lens and light source characteristics which are excluded during its measurement. [\[LINK\]](#)

## Chapter 3

### Software Design

In this project we built software that enables the user of our HS microscope configuration to control both the output wavelength of the light source and also the settings of the spectral imager in order to acquire clear spectral data. The designing procedure was divided into several tasks that were necessary for the completion of the software. These tasks are analyzed in the following paragraphs.

- Task 1:** Serial Communication with stepper motor
- Task 2:** Console application that moves the filter
- Task 3:** Create a GUI application for filter movement
- Task 4:** Embed the filter functionality to PGR FlyCap application
- Task 5:** Add more useful operations to the software

The division of the work to the previous tasks was in the spirit of bottom-up design. We start by implementing the most low-level (and essential) issue of this software design, the communication with the stepper motor. Then we test the communication we established by adding user control (in Tasks 2 and 3).

#### 3.1 Serial Communication with Stepper Motor

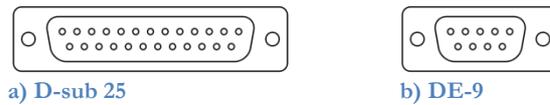
Computer serial ports most commonly use the RS-232 standard protocol. This is a standard for serial binary single-ended data and control signals connecting between a *DTE* (Data Terminal Equipment) and a *DCE* (Data Circuit-terminating Equipment). The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin out of connectors. An RS-232 port was once a standard feature of a personal computer for connections to modems, printers, mice, data storage, uninterruptible power supplies, and other peripheral devices. However, the limited transmission speed, relatively large voltage swing, and large standard connectors motivated development of the universal serial bus which has displaced RS-232 from most of its peripheral interface roles. Many modern personal computers have no RS-232 ports and must use an external converter to connect to older peripherals. Still the necessity of RS-232 communications remains because it is used in a wide range of scientific or industrial equipment, such as PLCs, VFDs, servo drives, and CNC equipment which are programmable via RS-232.

More specifically, in RS-232, user data is sent as a time-series of bits. Both synchronous and asynchronous transmissions are supported by the standard. In addition to the data circuits, the standard defines a number of control circuits used to manage the connection between the DTE and DCE. Each data or control circuit only operates in one direction, which is signaling from a DTE to the attached DCE or the reverse. Since transmit data and receive data are separate circuits, the interface can operate in a full duplex manner, supporting concurrent data flow in both directions. The standard does not define character framing within the data stream, or character encoding (such as ASCII or Unicode).

[\[LINK\]](#)

Also the RS-232 recommends, but does not make it mandatory, the use of the D-subminiature 25 pin connector (depicted in Figure 3.1.1a). Another very commonly used connector is the DE-9 pin connector (depicted in Figure 3.1.1b), which is also used in our project. The type of connector used

determines the pinout of the port that will be used. The pinout is the group of signal-to-pin assignments made in order for the DTE and the DCE to communicate.



a) D-sub 25  
b) DE-9  
Figure 3.1.1 Two commonly used serial port connectors

As aforementioned (see Chapter 2) in this project, the communication with the microcontroller of the stepper motor is made through a DE-9 cable which was customized in terms of cable connections between the two connectors' pinouts. Meaning that in the software controlling the serial port, we wouldn't have to control all the signals that are defined and used by the RS-232 standard. This customization makes the developers' life easier because he doesn't have to worry for any handshake or any other control implementation. The things that the serial port controller has to manage are two simple implementations of reading and writing data from/to the port.

Keeping that in mind, we studied how (in which language) we will communicate with the stepper motor. Our choice was to control the serial port through the Win32 API based on the fact that the software should be a Windows application.

Application programming interface (API) is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers. In the case of Windows 32-bit (Win32) API, the developer can use this set of methods in order to create an application that can interact with the Windows 32-bit operating system.

In Win32 API the interaction with almost any kind of object (file, port, device etc.) starts with the `CreateFile()` function. The returned/created object is manipulated through a `HANDLE` variable which is used as a pointer to the certain object. There are a lot of references and tutorials for serial communication in Win32 on the internet, so the process of creating a source code (a header file and a .c file) that establishes communication with the serial port. Our basic source of information was a System Services Technical Article from Denver Allen, 1995. [\[LINK\]](#) The source code created is written in C and compiled with Microsoft Visual Studio 2005 (MVS2005) and is used throughout the application stages described in the following paragraphs.

## 3.2 Console Application for Filter Movement

A console application is a computer program designed to be used via a text-only computer interface, such as a text terminal, the command line interface of some operating systems (Unix, DOS, etc.) or the text-based interface included with most Graphical User Interface (GUI) operating systems, such as the Win32 console in Microsoft Windows, the Terminal in Mac OS X, and `xterm` in Unix. A user typically interacts with a console application using only a keyboard and display screen, as opposed to GUI applications, which normally require the use of a mouse or other pointing device. [\[LINK\]](#)

The first step after establishing communication with the stepper motor via the serial port was to construct a simple console application (depicted in Figure 3.2.1) which will enable the user to control the movement of the stepper motor, ergo the filter movement.

```

c:\f:\documents and settings\administrator\my documents\visual studio 2005\projects\consoleDriver...
MENU:
1) Set DCB Settings
2) Start to move the motor
3) Exit
Give your selection: 1

DCB Settings OK!

Start moving the motor? (0/1): 1
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: 1
Status: Instruction Completed!
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: 2
Give the position: 500
Give the velocity: 2000
Status: Instruction Completed!
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: _

```

Figure 3.2.1 Console Application (first two commands)

As seen in the Figure above this application is very straight forward and simple. The user first has to set the DCB settings of the port in order to be sure that the communication will be successfully established. Then there are several options provided in a “MOVE MENU”. The first two options enable the user to move the filter either to the RESET position or to another specific one. The next two options are key operations of this project. Options 3) and 4) are the scanning instructions and their implementation is showed in Figure 3.2.2.

```

F:\Documents and Settings\Administrator\My Documents\Visual Studio 2005\Projects\consoleDriver...
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: 3
Give start position: 400
Give end position: 500
Give the step of the scan: 5
Give the velocity of each step: 2000
Give the timeout of each step: (milliseconds) 100
Scan Started...
Scan Completed...
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: 4
Give 1 position: 100
Give 2 position: 200
Give 3 position: 300
Give 4 position: 400
Give the timeout of each band: (milliseconds) 100
> 1 band
> 2 band
> 3 band
> 4 band
MOVE MENU:
1) Reset the motor to reference point
2) Move the motor to a position
3) Scan an area
4) Scan 4 positions (bands)
5) Go back
Give your selection: _

```

Figure 3.2.2 Console Application (two scan commands)

The scan instructions are classified in two types; “Scan Area” and “Scan Bands”. The user is asked to define five parameters for every scan. The “Scan Area” command starts a loop, instructing the stepper motor to move from a **start position** to an **end position** with a given **step**. The motor moves to each position with the given **speed** and remains in each position for a given amount of time (**time-step**). “Scan Bands” command follows almost the same implementation but instead of an area (range: start-end) it instructs the stepper motor to move to **four given positions** (bands) remaining in each position for **time-step** milliseconds.

Although the above presented application seems a bit ugly and not user-friendly it enabled us to define which instructions we need to implement in our GUI version of the application presented in the next paragraph.

### 3.3 Sample GUI Application

At this point of our design we wanted to integrate the previously analyzed functionality of a simple console application in a GUI based application. The first step was to transcend from an application that simply enables to move the stepper motor to an application that allows the user to move the mounted linear filter. Meaning, we had to move the stepper motor in terms of filter wavelengths (nm) and not motor positions. In order accomplish that we had to calibrate the stepper motor in accordance with the linear filter. The setup for this calibration was the following (Figure 3.3.1). Using a spectrometer (USB4000, Miniature Fiber Optic Spectrometer, Ocean Optics) we recorded the transmission spectrum of the linear filter for each position of the stepper motor. We moved the stepper motor using the above console application. The peak wavelength of each spectrum was registered in an Excel sheet as the emitting wavelength of the corresponding position. The resulted measurements were transferred into a .txt file which would be loaded to the GUI based application upon launching.

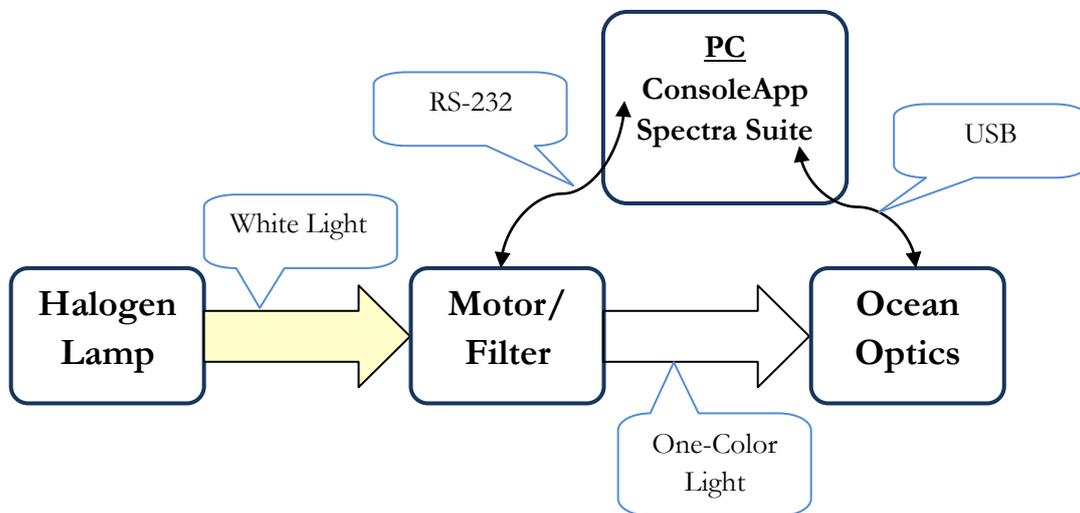


Figure 3.3.1 Stepper Motor Calibration setup

Although the wavelength range of the linear filter is from 400nm to 1000nm, only the 61.6% of it is used by the “calibrated” application. The effective filter wavelength range is from 426nm to 796nm. This range reduction is caused by the emission spectrum of the halogen lamp white light source (see Chapter 2). It is obvious that this calibration is filter and light source dependent. If it was to change the filter or the halogen lamp, it would be necessary to repeat the procedure.

As soon as we calibrated the stepper motor we started building the first GUI based version of this software. Since we wanted to create a “fast and dirty” prototype to test the GUI functionality, we constructed a Windows Application in C# (C-sharp). C# is one of the easiest and most versatile tools for creating GUI windows applications, which demand the minimum possible coding by the developer. The only drawback encountered was that C# is not compatible with the Win32 API that we used for the serial communication. There is a mechanism to bypass this problem called “pInvoke” [\[LINK\]](#); it is a Microsoft Visual Studio add-in which enables the translation from Win32 API to the C# API. So this C# application consists mainly of two classes; a class that translates the Win32 API and implements the functions of serial communication and a class that implements the functionalities (event handlers) of the GUI. These functionalities are analyzed here below.

Figure 3.3.2 shows how our first GUI application looks like, it is also depicted which are the main functionalities of the interface.

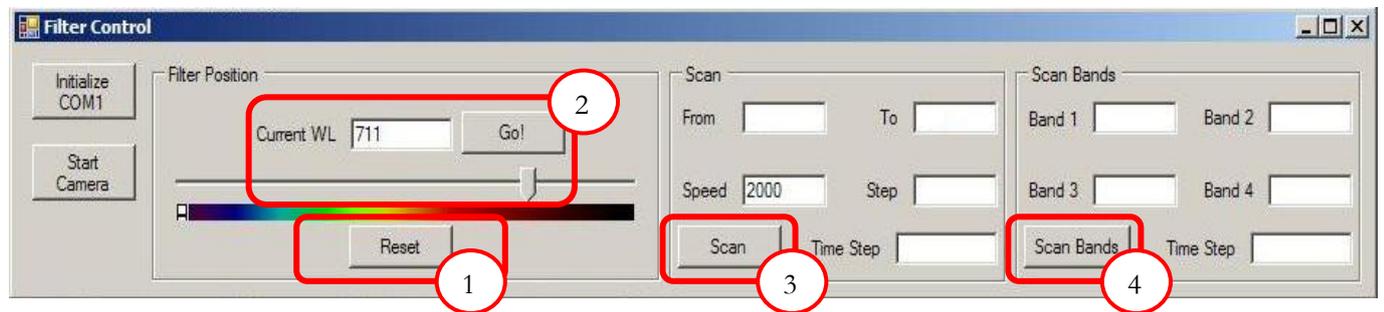


Figure 3.3.2 C# GUI application

### GUI Breakdown

- 1) **Reset Button.** The user has to press this button every time the application is launched in order for the RS-232 microcontroller to know the exact position of the stepper motor. When this button is clicked its' event handler calls a function defined in the serial class and sends the RESET instruction to the stepper motor.
- 2) **Filter movement.** This operation includes three graphical controls; the “Current WL” edit box, the “Go!” button and the wavelength slider. The user can move the filter either by typing the desired wavelength in the “Current WL” edit box and pressing the “Go!” button or by clicking on a position of the slider which will be translated in nanometers. Also if the slider “has focus” the filter can be moved by pressing the Arrow Keys (Right, Left). In both actions the two mechanisms are “synchronized” and if ones' value changes so does the others' (e.g. Figure 3.3.2 – 711 nm in the edit box → corresponding position of the slider). Both mechanisms are initialized (edit box value checking and sliders' min, max values) upon launching, when the stepper motor position Look-Up Table (LUT, calibration file) is loaded from the .txt file.
- 3) **Scan Button.** Pressing the “Scan” button will trigger the scan operation described (and implemented) in the previous paragraph. The user should fill the start (“From” edit box) and the end (“To” edit box) wavelengths of the scan, the step (in nanometers) of the scanning loop (“Step” edit box) and the “Time Step” (in milliseconds). The “Speed” edit box is initialized with the maximum value (2000) for convenience of the user, but it can be altered. If all parameters are correct the buttons' event handler calls repeatedly the move (write) function of the serial class with the corresponding positions (wavelength translated via the position LUT) to scan the given range of the filter.
- 4) **Scan Bands Button.** This operation is mainly the same with the “Scan” operation described above. The user should fill the bands (certain wavelengths) of the filter and the event handler will move the stepper motor to the corresponding positions waiting in each position for “Time Step” milliseconds.

The application analyzed above was the result of rapid prototyping procedure which led us to the final form of this software. The next step was to embed all above functionality to the PGR FlyCap software in order to integrate the tunable light source with a spectral camera.

### 3.4 FlyCap Embedded Application

The following section describes the most important and difficult part of the software design. In order to have a complete spectral imaging configuration (around a microscope) we need to control a spectral imager simultaneously with the tunable light source. Hence the notion was to embed a graphic control (a toolbar or a dialog window) with the shape, form and functionality of the GUI described above in a software application that controls a spectral camera. Since we have selected DragonFly to be the spectral imager we have to modify the corresponding software of PGR which controls PGR camera; FlyCapture 1.6 (FlyCap) (Figure 3.4.1).



Figure 3.4.1 PGR FlyCapture v1.6, main view

Before analyzing our extension to this software we have to mention some key information about FlyCap and its architecture. This software is a Microsoft Windows application, written in C++, and uses the Microsoft Foundation Classes (MFC). MFC wraps a big part of the structures of Win32 API (windows.h) in to C++ classes. It is created to make developer's life easier in constructing Windows applications, since the Win32 API is very complex. So MFC saves the developer time and effort by automating some structures that are necessary for a GUI application and makes the coding a little easier. Lastly MFC provides, via a project creation Wizard, a variety of working GUI templates in order to minimize the “built-from-scratch” applications which demand more labour and careful designing.

FlyCap uses one of the MFC provided templates, the Single Document Interface (SDI). SDI applications allow only one open document frame window at a time. The developers of PGR have embedded the viewing of the camera in the main frame window. Using also their own API with functions and classes to control both the camera and the GUI they constructed an application which is complex and powerful in terms of camera control (Figure 3.4.2 depicts the settings interface of the camera).

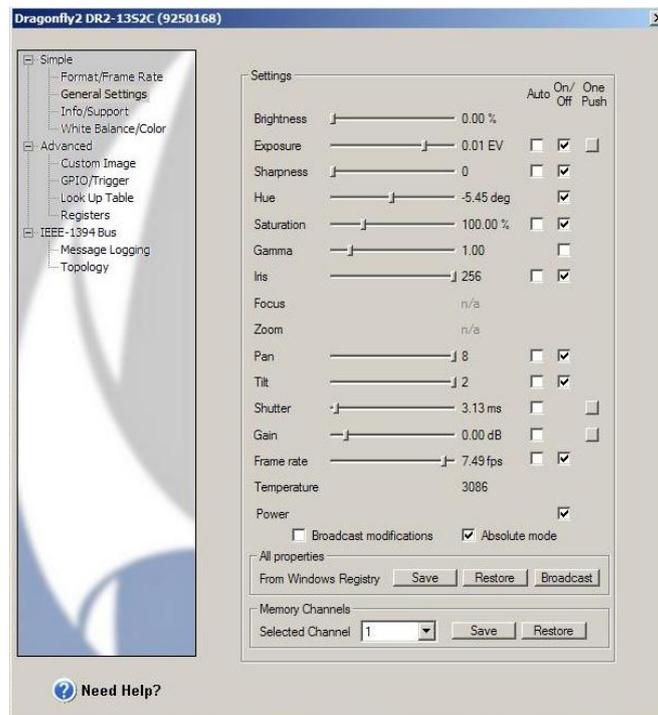


Figure 3.4.2 PGR FlyCapture v1.6, camera settings interface

This is where our work comes in, to expand the control of this software. In order to do that we created and added a number of classes and structures on top of the FlyCap source code. We call this bundle of classes TUCFlyCap. The structure and the interconnection of those classes are analyzed in this paragraph. But, once again, before we go on to specific descriptions of TUCFlyCap architecture lets concentrate a bit in some important and necessary mechanisms/structures of a (Microsoft Windows) GUI application.

- a) **Messaging**, the backbone of every GUI application. This structure is used to transfer information and notifications throughout the applications' classes concerning an event (e.g. a button of the GUI is pressed). Messages are used basically for any kind of intercommunication in the application.
  - i. **Message Loop**, in every GUI application there is a message loop that is used to service all the messages from the message queue and notify the appropriate class. It is essential that this loop is never blocked or else the GUI becomes unresponsive.
  - ii. **Message Handlers**, when a class is meant to transact using messages it has to have as member functions some message handler. These functions are the ones called from the message loop when an "interesting" message comes up. The most common example for message handlers is the button handlers which define the actions made upon a button press.
  - iii. **User-defined Messages**, the mechanism of messages is very useful so it is almost standard that a GUI application has its own messages defined by the developer in order to extent the class intercommunication.

The aforementioned Message Loop is one of the structures that are automated in MFC. When programming in Win32 a developer would have to write his own loop, but this taken care of in MFC. The developer only needs to declare the message handlers; this group of declarations is called the MESSAGE\_MAP.

- b) **Threads**, a well-known tool! We will not elaborate very much on threads here but it is something I have to mention because our software uses multi-threading.
- i. **User Interface (UI) Threads**, these are the threads that are created from a class of the application in order to implement another GUI class. By definition these UI threads are connected to a CWnd object, which is the main MFC GUI class. So if for some reason it is wanted for the execution to continue in a separate thread and also user interaction is needed we create a UI Thread.
  - ii. **Worker Threads**, these threads are simpler, they are only designed to do “background” work with no interaction through messages. To create a worker thread we only need to define a thread controlling function.
- c) **Main MFC classes**, in order to understand in depth the following analysis of the TUCFlyCap architecture it is essential to describe some of the main classes of the MFC library, which are also used in our software.
- i. **CWnd**, a CWnd object is distinct from a Windows window, but the two are tightly linked. A CWnd object is created or destroyed by the CWnd constructor and destructor. The Windows window, on the other hand, is a data structure internal to Windows that is created by a Create member function and destroyed by the CWnd virtual destructor. The DestroyWindow function destroys the Windows window without destroying the object. The CWnd class and the message-map mechanism hide the WndProc function.
  - ii. **CDialog**, the base class used for displaying dialog boxes on the screen. Dialog boxes are of two types: modal and modeless. A modal dialog box must be closed by the user before the application continues. A modeless dialog box allows the user to display the dialog box and return to another task without canceling or removing the dialog box. A CDialog object is a combination of a dialog template and a CDialog-derived class. A dialog box, like any other window, receives messages from Windows. In a dialog box, you are particularly interested in handling notification messages from the dialog box's controls since that is how the user interacts with your dialog box.
  - iii. **CDocument**, a document represents the unit of data that the user typically opens with the File Open command and saves with the File Save command. CDocument supports standard operations such as creating a document, loading it, and saving it. The framework manipulates documents using the interface defined by CDocument. An application can support more than one type of document; for example, an application might support both spreadsheets and text documents. Each type of document has an associated document template; the document template specifies what resources (for example, menu, icon, or accelerator table) are used for that type of document. Users interact with a document through the CView object(s) associated with it.
  - iv. **CView**, a view is attached to a document and acts as an intermediary between the document and the user: the view renders an image of the document on the screen or printer and interprets user input as operations upon the document. A view is a child of a frame window. More than one view can share a frame window, as in the case of a splitter window. The relationship between a view class, a frame window class, and a document class is established by a CDocTemplate object.

After understanding the above mechanisms we will elaborate on a general overview of the TUCFlyCap architecture.

## General Overview

Now we can take a deeper look at the TUCFlyCap architecture. Firstly, we have to note that the main functionality (that also interests us) of the FlyCap software is implemented in the CFlyCapDoc class, which is CDocument derived class. This contains the implementation of all the menu items handlers and also implements the grabbing of the images (grab loop) from the camera. Based on the CFlyCapDoc implementation we added some menu items and their handlers.

When the user chooses, from the menu items, to manipulate the filter we need to create a new instance of a CDialog derived class that provides to the user the necessary GUI options, the same functionality with the previously presented C# GUI application. This class is the FilterDialog class that implements all the functionality to move and scan the filter. The scanning procedure is a looping procedure, and it affects/blocks the constantly running message loop. A single thread cannot execute simultaneously two loops. Even if the implementation allows the execution to “switch” from one loop to another, in our case there is another problem. As we mentioned the CFlyCapDoc object runs another loop, the grab loop. So a scan procedure executed by the FilterDialog object affects the grab loop of CFlyCapDoc object, therefore the real time viewing of the images from the camera. Hence, multi-threading was the only way to disentangle the two loops (grab and scan loop). We created a UI thread (FilterThread) that is connected with an instance of FilterDialog. This was basically the only step we had to take in order to enable the filter control embedded in the FlyCap application.

Another issue we had to solve was the communication of our software with the spectral camera. In order to acquire clear images (spectra) it is essential that the camera changes its settings according to the excitation wavelength, meaning according to the stepper motor’s position. The main settings of the camera that are usually altered according to the excitation are the shutter and the gain. In other words it is essential that the camera is calibrated in terms of settings according to the filter movement. This calibration is applied step by step. There is no way for the camera to hold the corresponding values of the settings for each wavelength of the filter nor to keep track of the position of the stepper motor and apply automatically the desired settings. Inevitably the calibrated settings have to be controlled by an object that has a calibrated group of settings loaded and can monitor the movement of the filter. Our first guess was to let FilterDialog to be that object.

After studying a part of the FlyCap code we found out that the camera control desired (change camera settings) is implemented in the CFlyCapDoc class. The first notion was to understand this implementation and try to apply something very similar in a part of the FilterDialog class. We discovered that all the changes instructed to the camera are made through a member variable of CFlyCapDoc which is FlyCaptureContext (FCC) type. This type is defined in the FlyCapture API that is used from PGR. Our first attempt to gain control over the camera through this type of variable was to pass it to the FilterDialog class by reference. But, as maybe expected, this attempt did not work out, because of threading/resource problems. We could not use the reference of the FCC variable from another thread neither we could create a new FCC variable, because we would then need to use a great deal of initialization code of CFlyCapDoc which is based on PGR’s API and not well documented.

The second idea we attempted was to notify somehow the CFlyCapDoc class that we want to apply changes on the camera. As we mentioned before the notifications among the classes is done using messages (system or user-defined). When programming the FilterDialog to send a user-defined message every time camera control was needed we realized that it is not possible for a CDocument (CFlyCapDoc) instance to receive and process user-defined messages. In the concept of an SDI all user-defined messages are received from a CView (which is a subclass of CWnd) derived object and most of the control actions are done by a CDocument object. So as user-defined messages were our

only way to go and since CFlyCapDoc was unable to process those messages we implemented an oddly popular solution which is called Message-only Window. We created an invisible to the user CWnd derived class which communicates by reference with CFlyCapDoc (this is possible since both classes are executed in the same thread) and receives user-defined messages from the FilterDialog class. This class was named for obvious reasons SecretControlWnd.

As we implemented this message-only class it became clear that in order to save/capture an image we needed to have at any time at our disposal the image (raw data) grabbed from the camera. We tried to modify the implemented (in CFlyCapDoc class) grabbing loop, so that every time a new frame came into “process” from CFlyCapDoc a variable in the SecretControlWnd was updated to that new frame. Also after testing this inter-class frame update we realized that in order for the settings changes or the filter movement to complete and to be depicted correctly in the image we would have to wait a number of frames. Waiting every time before save (or measuring reflectance for calibration) three frames guarantees that the frame saved is the correct one, but also adds a minimum delay in the system’s performance which depends on the frames-per-second (fps) that the camera sends to the PC.

The last main functionality wanted for our GUI extension is an automated calibration procedure. The calibration is supposed to measure the average reflectance (value of pixel in gray scale [0-255]) of a small central of the current frame and accordingly change the settings (shutter, gain) of the camera in order to succeed reflectance of a wanted (user input) value. To implement the above procedure we created a another CDialog derived class, CCalibrationDialog, which in collaboration with SecretControlWnd calculates a pair of camera settings for every position of the filter (wavelength) so we get all along the length of the filter the same reflectance. This automated calibration procedure will be analyzed further in this chapter.

All the structures and classes described in the above seven paragraphs implement the biggest part of our software architecture. Hopefully the next Figure will make things clear:

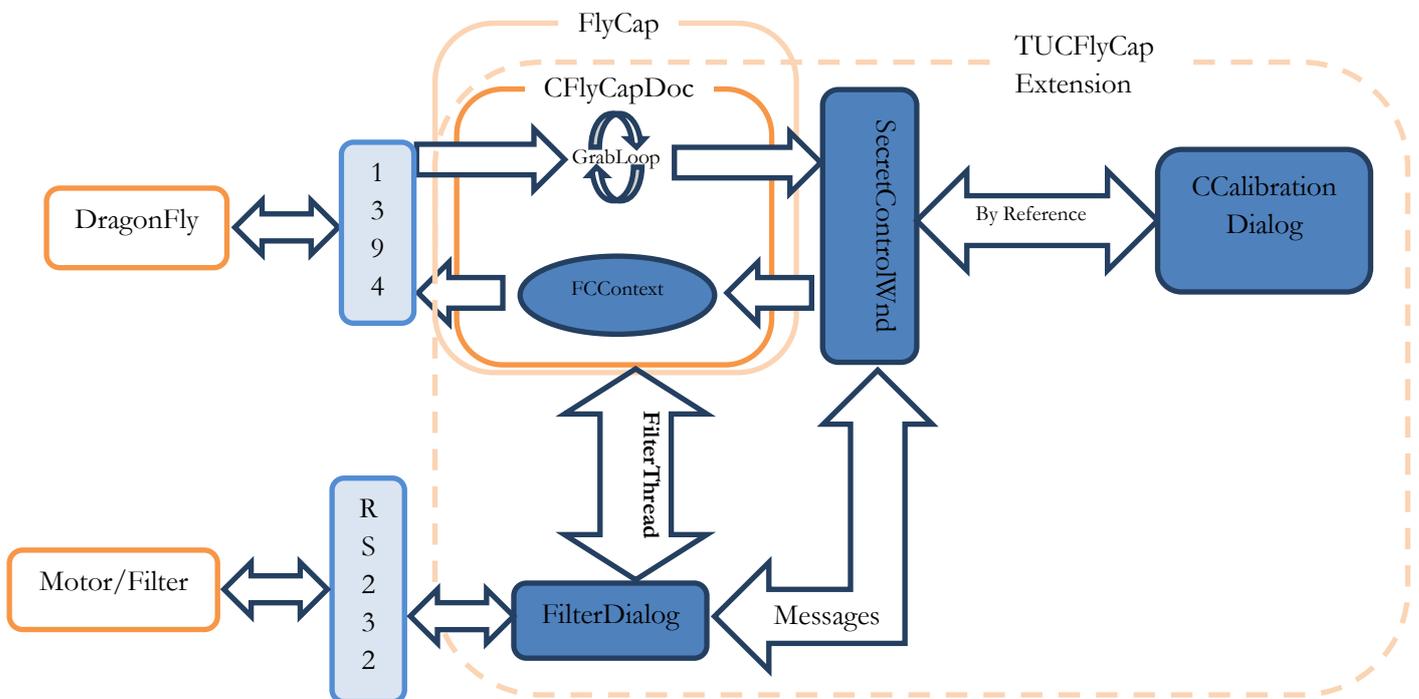


Figure 3.4.3 TUCFlyCap General Overview

## GUI breakdown

After describing the main architecture issues and characteristics of the TUCFlyCap software, we will analyze its functionality from the user perspective. In the following series of figures and descriptions we will elaborate on the possible user actions.

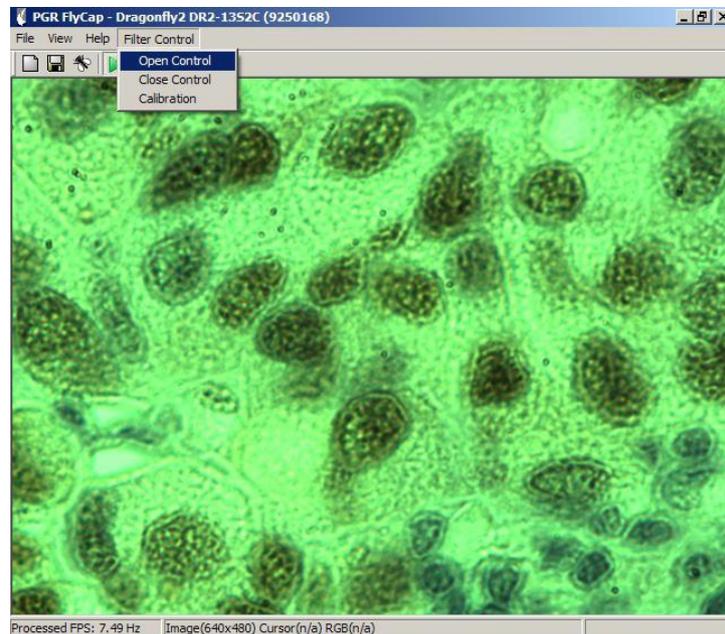


Figure 3.4.4 “Open Control” action

When the user selects the “Open Control” option under the “Filter Control” menu item, then a new dialog is popped which implements the functionality of filter control (Figure 3.4.5).

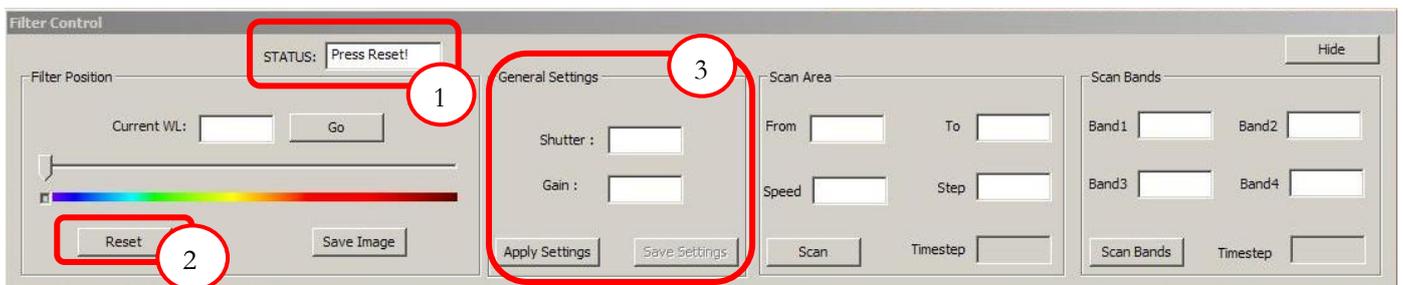


Figure 3.4.5 Filter Control (Upon opening)

- 1) **Status Edit Box**, when the user first opens the control, he is instructed by this edit box to press “Reset” button. This edit box gives feedback to the user throughout the actions he attempts on this control dialog.
- 2) **Reset Button**, same functionality as in 3.3.GUI Breakdown. When the user presses this button then the filter control dialog is ready to execute any other instruction (movement, scan) (Figure 3.4.6).
- 3) **General Settings Frame**, this frame displays the current settings (shutter, gain) of the camera according the filter movement. In order for this frame to be active it necessary that calibration is loaded. If not the two edit boxes display “N/A”, but the user can enter two values and then test them on the image by clicking “Apply Settings”. The functionality of this frame will be analyzed furthermore in the next paragraph.

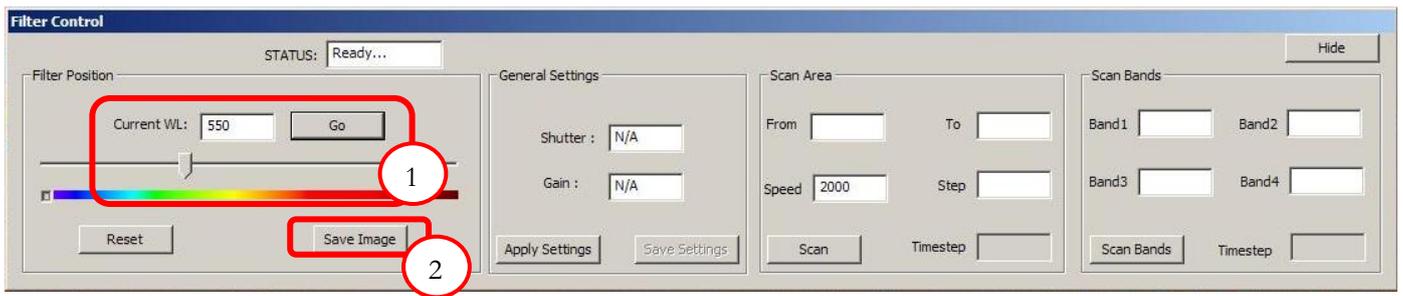


Figure 3.4.6 Filter Control (Moved to 550nm)

- 1) **Filter Movement**, the filter is moved to a desired wavelength (550nm) either by “Go” button or by clicking on the wavelength slider (see 3.3.GUI Breakdown). The results<sup>1</sup> of this action (and other movements are shown in Figure 3.4.7).
- 2) **“Save Image” Button**, the user can move the filter to any wavelength and he can choose to save the current image. The image is automatically saved in a folder called “Sessions/Session\_dd\_mm\_yyyy-hh\_mm\_ss”. The main folder “Sessions” is automatically created upon the **first** launch of TUCFlyCap, in the current working directory (meaning the path of the executable). Every time application is re-launched the software creates a subfolder with the prefix “Session” and the current date (dd\_mm\_yyyy) and time (hh\_mm\_ss). This way the user knows exactly where the images of each session are saved. The image names are also automated. If an image is saved by the “Save Image” button then the storing path is “rand/calc\_i\_www”<sup>2</sup>. The subfolder “rand” is again automatically created upon launching of the application and it is used to store all the images that are saved by the “Save Image” button (non-scanning procedure). The image name has the prefix “calc” and a counter “i” as well as the wavelength of the image “www”. The structure of the folders and the images saved are depicted in Figure 3.4.8

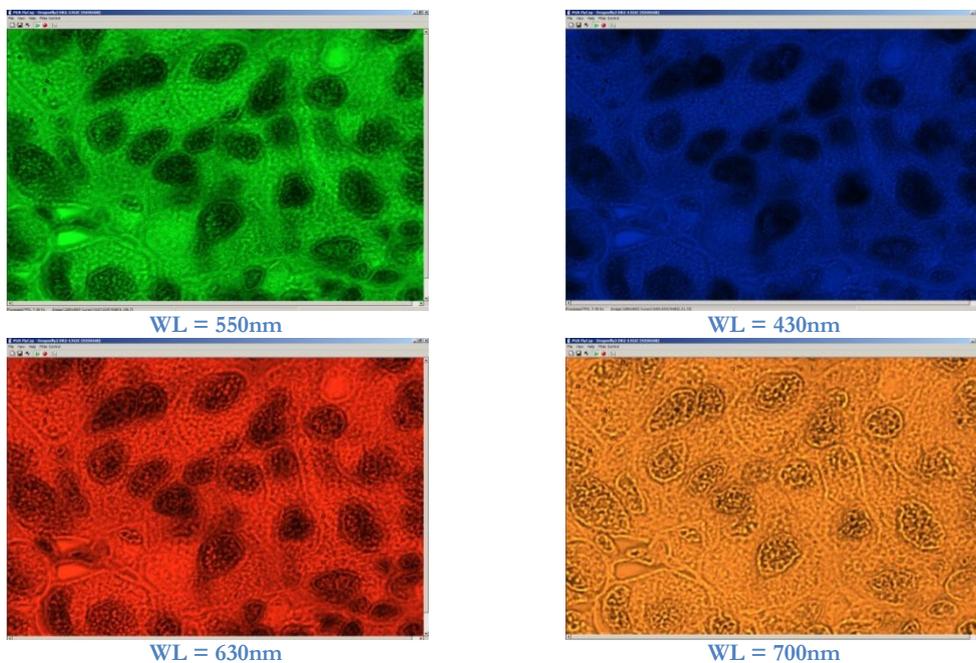


Figure 3.4.7 Camera viewing for various wavelengths of the filter

<sup>1</sup> Although the microscope system will be operated in gray scale mode, the images are shown in RGB mode for ease of presentation.

<sup>2</sup> The images stored from scans are a different case and they’re analyzed further below (scanning → “cubes”)



Figure 3.4.8 Image storing path

The next figure shows the state of the filter dialog when a “Scan” command is in execution. The viewing of the camera is not shown for ease of presentation.

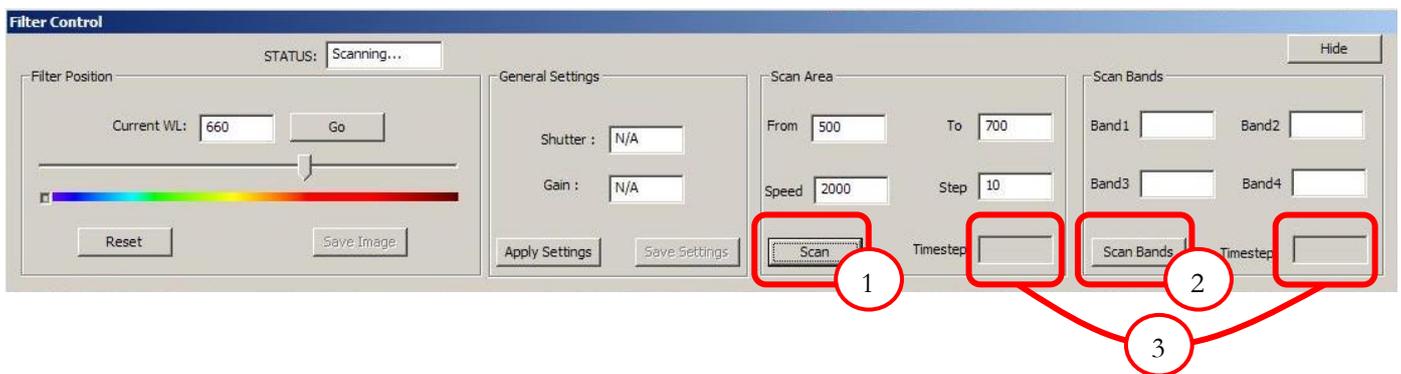


Figure 3.4.9 Filter Control (Scanning procedure)

- 1) **“Scan” Button**, when the user presses this button, the application firstly checks the values inserted (“From”, “To” and “Step” edit box) and if valid it launches a scan procedure. Every time the filter moves to a new position the Filter Position frame (“Current WL” edit box and wavelength slider values) is updated.
- 2) **“Scan Bands” Button**, the handling of this button is similar to the one described above.
- 3) **“Timestep” Edit Box**, as depicted these two edit boxes are disabled. The scanning procedure is slightly changed in this version of our software. We implemented a more efficient way of scanning. This procedure is described in the next section.

### Scanning Loops

When a scan operation (“Scan”, “Scan Bands”) is triggered, a loop starts executing in the FilterDialog class. The FilterDialog instance sets the (next) position of the filter and then it enters a waiting loop. As soon as the filter completes the moving instruction the SecretControlWnd instance is notified by reading from the serial port<sup>1</sup>. On notification the SecretControlWnd instance starts counting three frames (necessary waiting frames for the image to update to the new conditions [settings, excitation]) and when this process is completed it saves the current image of the scan. As a

<sup>1</sup> This serial reading from SecretControlWnd instance is analyzed furthermore on this chapter.

last step the message-only window notifies through a user-defined message the FilterDialog instance that the image is saved and the loop continues to the next position of the filter. The following figure depicts this procedure.

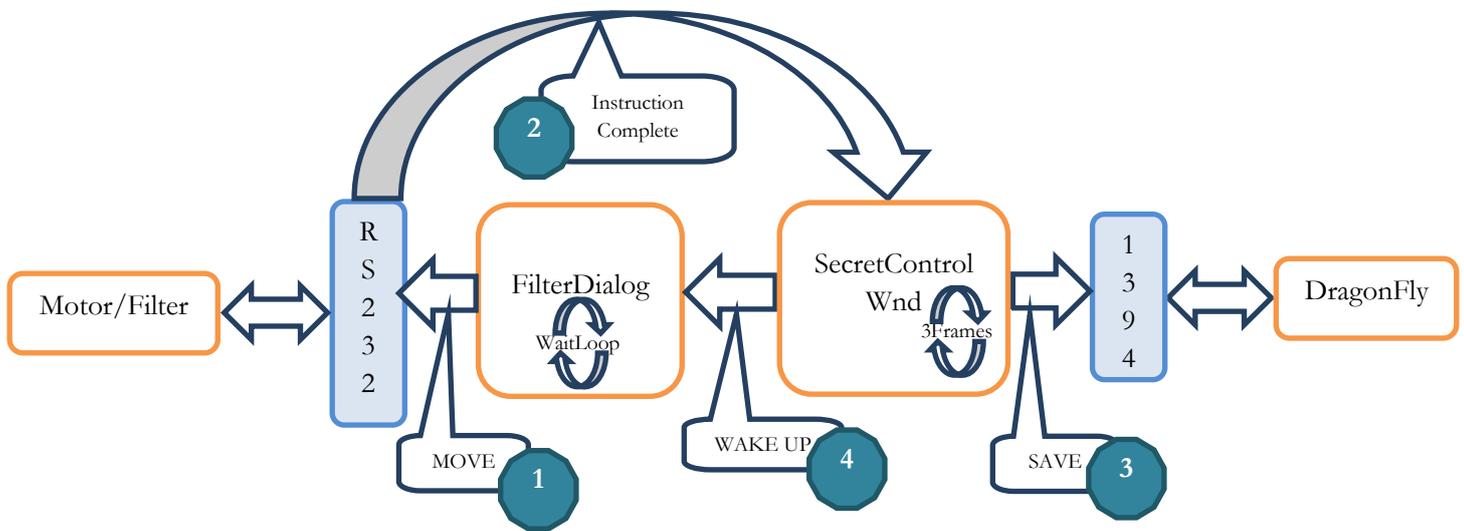


Figure 3.4.10 Scanning procedure

### Saving a “Scan” image

When a “Scan” image is stored its path is defined by a similar procedure to the one of the “Save Image” button procedure. As depicted in Figure 3.4.8 the special subfolder for “Scan” images is called “cubes”, this is because scans of the filter are the commands to acquire spectral cubes. The counter “i” in the name of each image remains the same for every image of the same cube. So the counter is augmented every time the “Scan” button or the “Scan Bands” button is pressed. This way the images can be grouped as spectral cubes.

### Reading from Serial Port

As we mentioned earlier, it was necessary for the completion of this project to implement an efficient control over the Serial Port. Also we mentioned that the SecretControlWnd instance was waiting for the instruction of the motor/filter to be complete. When it was triggered it finished the procedure for one loop of our scan. This way there are no delays inserted by the user and the system always uses its maximum scanning speed (which is **frame-rate** dependent).

But how does the SecretControlWnd waits for the completion of the filter instruction? It reads the Serial Port. This solution is a combination of a worker thread and the implementation of our Serial control. We created a function that, until the handle of the serial is deleted, reads from the port. When a succeeded read occurs it changes a public flag of the Serial class denoting that event. The aforementioned function is basically the control function of a worker thread created from SecretControlWnd instance when the application is launched. The worker/reader thread continuously reads from Serial Port and every time an instruction is completed changes the flag. This thread implementation is a very efficient one since the reading is done as a background work and it does not delay other procedures managed by SecretControlWnd.

Lastly we need to clarify another issue, how the SecretControlWnd instance has access to the Serial Port. As we saw in this chapter the one class dealing with the communication with the Serial Port

is FilterDialog. This class creates, initializes and writes to the Serial Port. But as soon as the serial object is initialized then its handle is passed through a user-defined message to the SecretControlWnd. Then the worker thread is created and the public “completion” flag of the serial class can be checked by SecretControlWnd.

This section was dedicated to the main functionality of our software TUCFlyCap. The last part of this chapter will be a paragraph that describes the automated calibration operation by an algorithmic and a user perspective.

### 3.5 Automated Calibration Operation

In the previous section we saw that it is essential for the camera to be calibrated according to the excitation wavelength that is provided from the tunable light source. Also, as aforementioned, the calibrated settings should be provided to the camera from the TUCFlyCap software, since the camera has no capability of storing those settings or keeping track of the filter position. So the calibration should either loaded from a file into the application or created from the application itself. Manual calibration of 370 positions (wavelengths 426nm to 796nm) of the filter is a very long and laborious task. Therefore we created an automated procedure that creates a calibration file and fills it with the appropriate settings for each position of the filter.

The idea was to measure/calculate the average value of some central pixels of the image to get an average reflectance value. Then the application would try to change the settings of the camera and recalculate the reflectance until its value matches a given reference reflectance value given by the user. If these two values match, then the software would move the filter to the next position and repeat the algorithm. Also the user has to fill the wavelength step of the procedure. The implementation of the automated calibration allows a step (gap) between calibrated wavelengths for faster completion of the procedure. The wavelengths that are not calibrated from the algorithm are filled with linearly fitted values according the previous and next positions.

This general view of the algorithm provokes some objections; the reflectance of some central pixels does represent well the average reflectance of the whole picture and if the wavelength step is “big enough” then the fitted values will not approach an acceptable calibration. There are more efficient ways to calibrate a camera [\[LINK\]](#) that will not present these drawbacks, but the efficiency of the calibration algorithm is not an issue of this thesis. Beside the drawbacks mentioned the method we constructed works quite well and we will elaborate on some details here below.

The calibration procedure starts by setting the gain of the camera to zero. Then it enters an “endless” loop in which it increases/decreases the camera settings according to the measured average reflectance. The newly calculated settings are applied to the camera and the reflectance is again calculated. The loop terminates when the reflectance is (almost  $\{\pm 0.5\%$ ) equal to the desired reflectance value. Before the function exits it writes the calibrated pair of settings to the settings LUT, which is a member variable for inner storage of the calibration. Upon exit, this LUT is saved to a selected .txt file.

The increase/decrease of the settings depends on the current measured reflectance. The weight (percentage) of the increase/decrease that is applied to the settings changes according to the difference

between the current and the desired reflectance. Every time the function applies new settings these weights are updated. Also the function first tries to reach the desired value by changing only the shutter of the camera. If the shutter changing fails to reach that value then the function tries by changing the gain. The weight selection is depicted in the next figure.

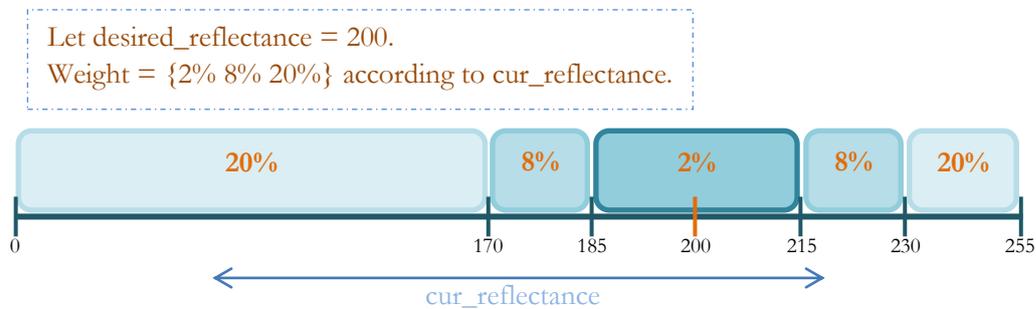


Figure 3.5.1 Weight selection

For every zone presented above, there is a counter that counts how many times the function tried to approach the desired reflectance with each weight. If that number exceeds a predefined (relatively high) limit that means that the algorithm is oscillating back and forth to same weight zone. This is solved by forcing the weight to the next lower zone. In the worst case, where the oscillation is in the lowest zone (2%), then the change of the value of the setting (shutter/gain) becomes accumulative (e.g. shutter = shutter ± MIN\_STEP). MIN\_STEP is a predefined constant which is the smallest possible change to apply to the settings (it is different for shutter and gain!).

The second part of this section will be the analysis of the user perspective of this procedure.

### GUI Breakdown

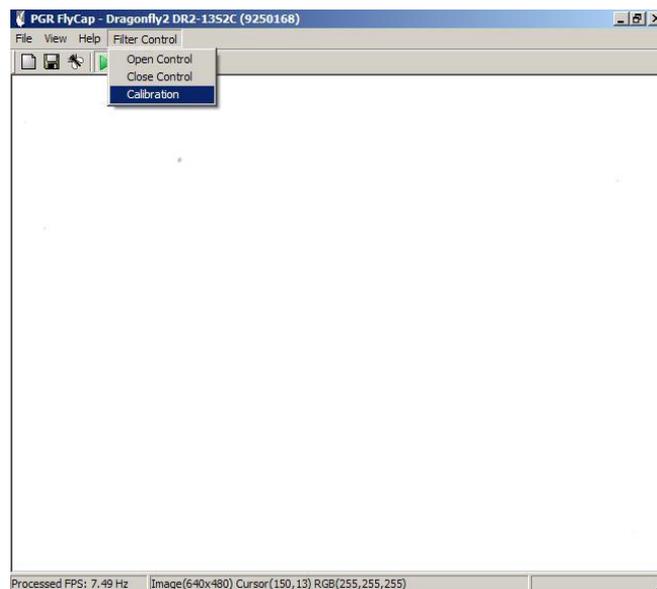


Figure 3.5.2 Calibration selection

When the user selects the “Calibration” option under the “Filter Control” menu item a new instance of the CCalibrationDialog is created and its dialog pops (Figure 3.5.3).

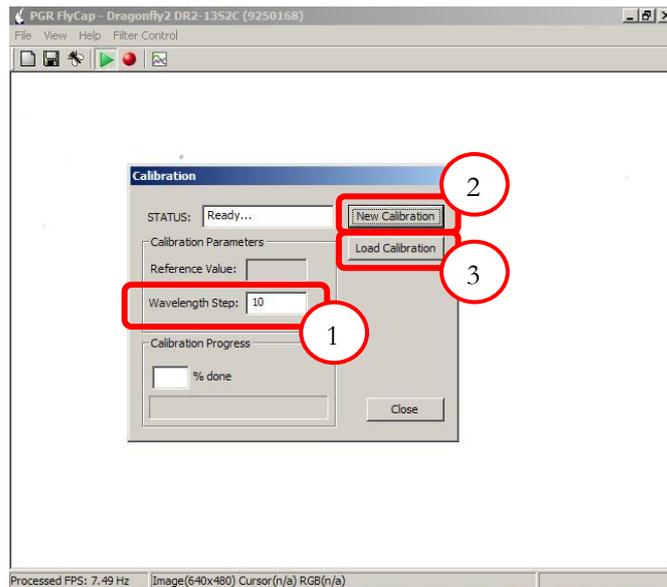


Figure 3.5.3 Calibration Dialog (upon launching)

- 1) **Wavelength Step**, if the user wants to create a new calibration, first action is to fill in the step edit box.
- 2) **New Calibration Button**, when the user fills in the “wavelength Step” edit box he can proceed to start a new calibration procedure. (Figure 3.5.4)
- 3) **Load Calibration Button**, if the user wishes to load a calibration file and not create a new one then he should select the “Load Calibration” button. When pressed this button creates a new CFileDialog instance (standard modal dialog of MFC) which allows him to browse in the file system and select a .txt file to load.

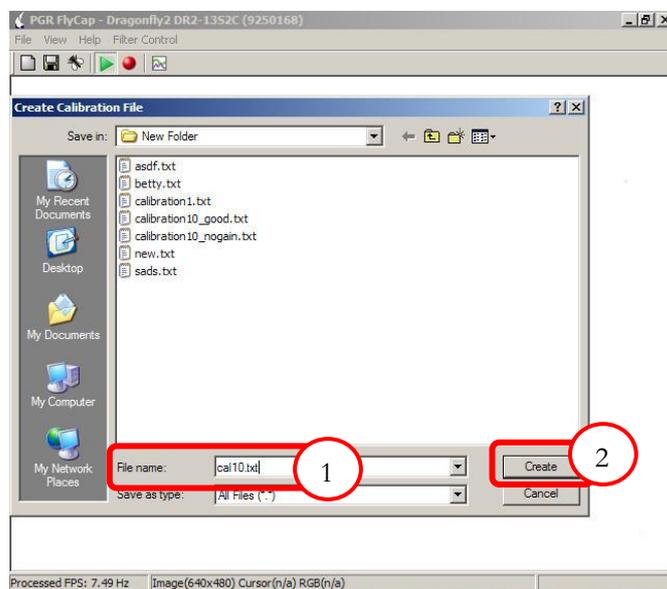


Figure 3.5.4 File Dialog to create the new calibration file

- 1) **File name**, the user fills in the wanted filename of the new calibration .txt file.
- 2) **Create Button**, when the user presses the “Create” button a new instance of CReferenceDialog (dialog depicted in Figure 3.5.5) is created and pops in order to execute the last step of this user interaction procedure.

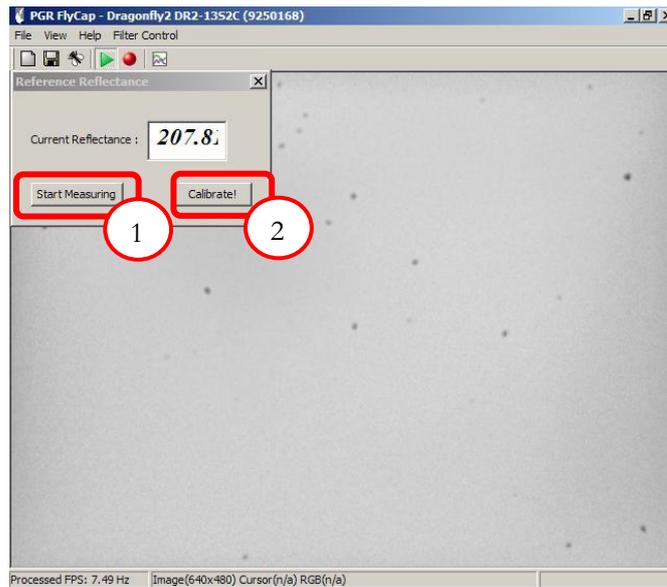


Figure 3.5.5 Reflectance Dialog

- 1) **Start Measuring Button**, when this button is pressed the dialog displays the current average reflectance of image. The user then can adjust the iris and the shutter of the camera as well as the excitation wavelength from Filter Control dialog in order to achieve a desired value of reflectance on a specific wavelength.
- 2) **Calibrate Button**, when the user achieves its wanted reference reflectance he presses this button and the automated procedure starts (Figure 3.5.6).

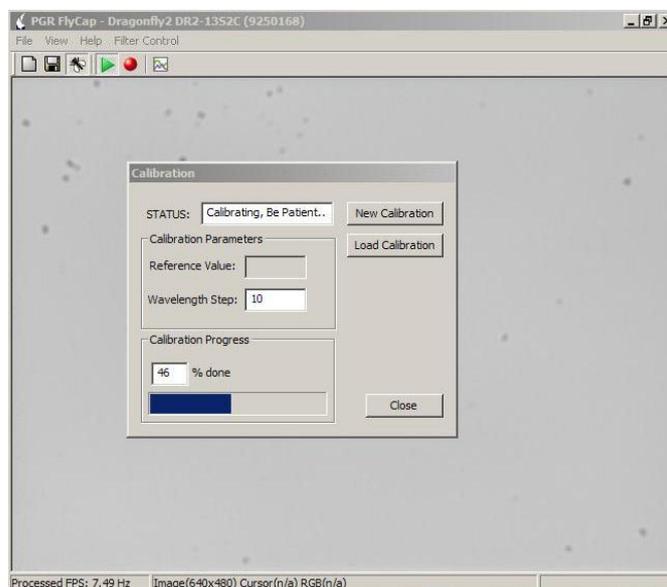


Figure 3.5.6 Automated Calibration Procedure

When the procedure is complete then the user can move the filter and do scanning loops but now the settings given of the camera will be calibrated according the filter wavelength and the General Settings frame will be fully enabled.

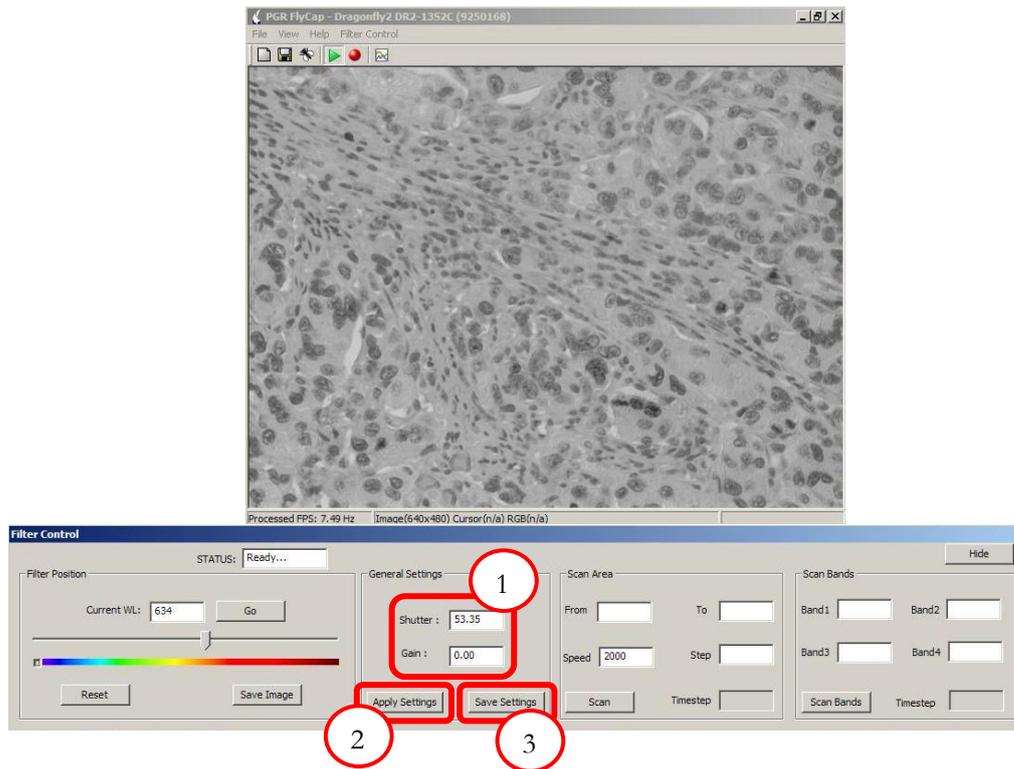


Figure 3.5.7 Filter Control with completed Calibration

- 1) **General Settings Edit Boxes**, here the calibrated values of the settings of the camera are displayed according to the filter position.
- 2) **Apply Settings Button**, the user has the option of changing the shutter and gain of the camera and applying them on the current image viewed. If the user does not press “Save Settings” button then the values that he filled in are lost.
- 3) **Save Settings Button**, if the user wants to save some values of the settings and wants to change the calibration file created or loaded then he presses this button. The next time that the filter visits the position he altered, the values of the settings will be the ones he saved.

## Calibration Results

In the previous section we fully described the automated calibration procedure, but we need a way to confirm that the results (calibrated camera settings) are accurate and that the calibration produced is satisfying enough to substitute the manual calibration procedure. The next figure shows the results of a spectral cube analysis tool constructed by George Epitropou as a part of his Diploma Thesis [\[LINK\]](#). We used this tool to confirm the results of our calibration.

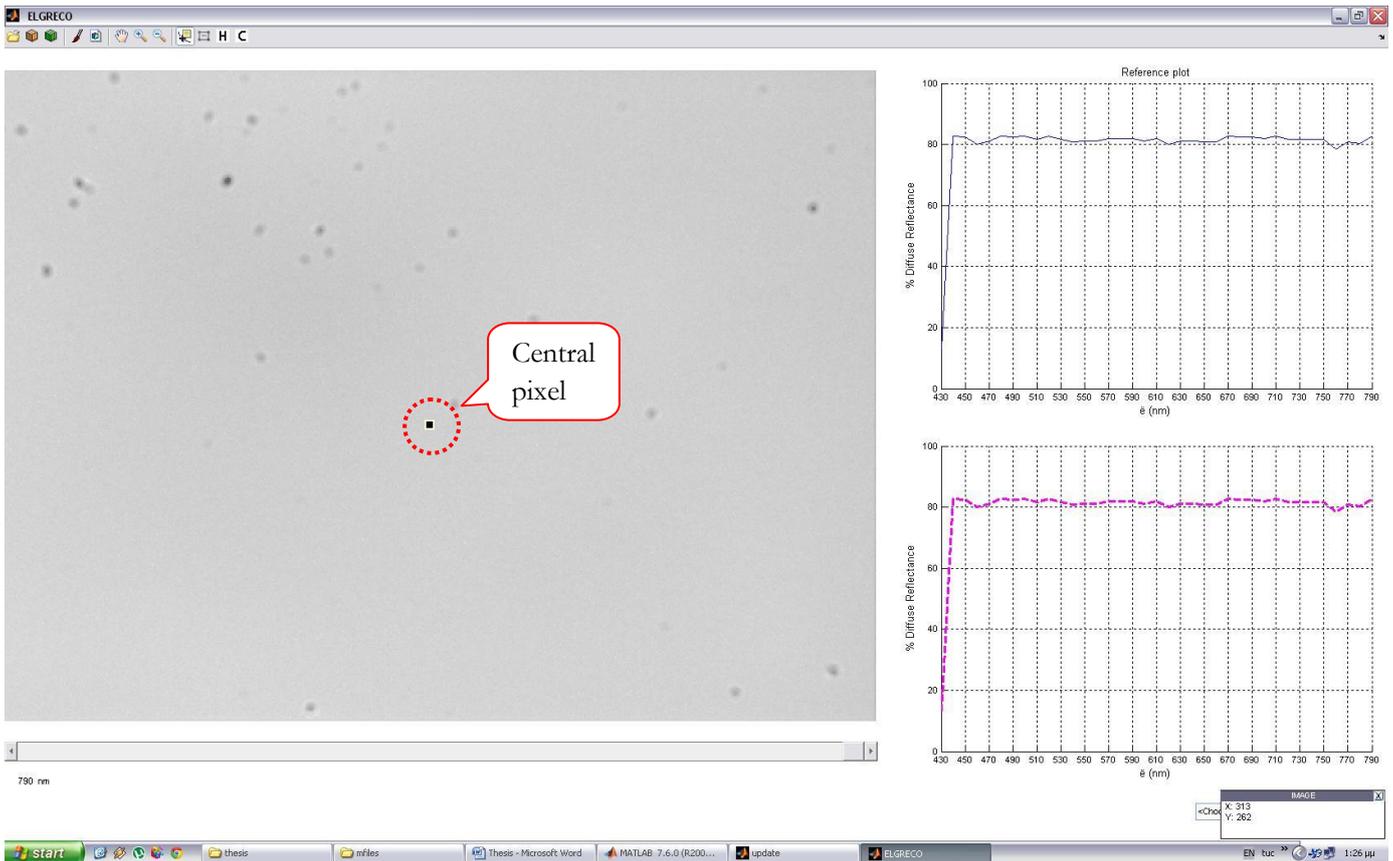


Figure 3.5.8 Calibration results confirmed with ELGRECO tool

Firstly, we acquired a spectral cube from the TUCFlyCap application, after completing a “New Calibration” procedure as shown above. The cube was acquired by executing a scan loop from 430nm to 790nm with step 10nm. There were no specimens in between the excitation light source and the camera. Secondly, the resulted images were given as input to the ELGRECO tool. This tool enables the user to observe a pixel’s value in every “slice” of the spectral cube simultaneously, by presenting these two plots shown on the right-hand side. When only one pixel is selected (the case above) the two plots are identical and they represent the percentage of Diffuse Reflectance. As shown the reflectance values of a central pixel remain in the same level (with a very small variance) almost in every wavelength. There is an incapability of succeeding a good calibration in the “low-blue” region of the filter, where the emission of the halogen lamp is very low.

The results presented above are for a calibration created with wavelength step of 10. The results of the algorithm do not improve even if the procedure calibrated every possible position of the filter (wavelength step == 1). Also the images inserted to ELGRECO tool were not directly calibrated but linearly fitted by the algorithm. The results show that this fitting produces very satisfying results if the

step is not over 30 nm! Lastly we saw that the directly calibrated images and the fitted ones do not present significant differences.

## Chapter 4

### System Evaluation and Performance

In this Chapter we will elaborate on the performance of the system we built. The aspects of evaluation will be the accuracy of the bipolar stepper motor and thus the center wavelength of the LVF, the output light power that is provided as input to the microscope and the frame rate of our hyper-spectral imaging system when scanning across the length of the LVF (acquiring spectral cubes). All the aforementioned parameters are important to define the stability and quality of operation of a spectral imaging system.

#### 4.1 Accuracy of Stepper Motor

One of most crucial parts of our design is the movement of the LVF in order to tune the output light's wavelength. This movement is caused by the bipolar stepper motor we use and the translation from stepping positions to center wavelengths on the LVF is made through a calibration that we have done used by the driving software/GUI. In order for our system to be consistent it is necessary that the stepper motor is very accurate in stepping to a certain position/wavelength. A probable lack in accuracy would mean that our system does not repeatedly and consistently produces the same output light given the same instruction (desired wavelength). So it is essential to test how accurately the motor moves. In the next figure we present a series of measurements of the distance the motor traversed in (mm) by moving for the same amount of steps (200). The measurements were made using a digital caliper.

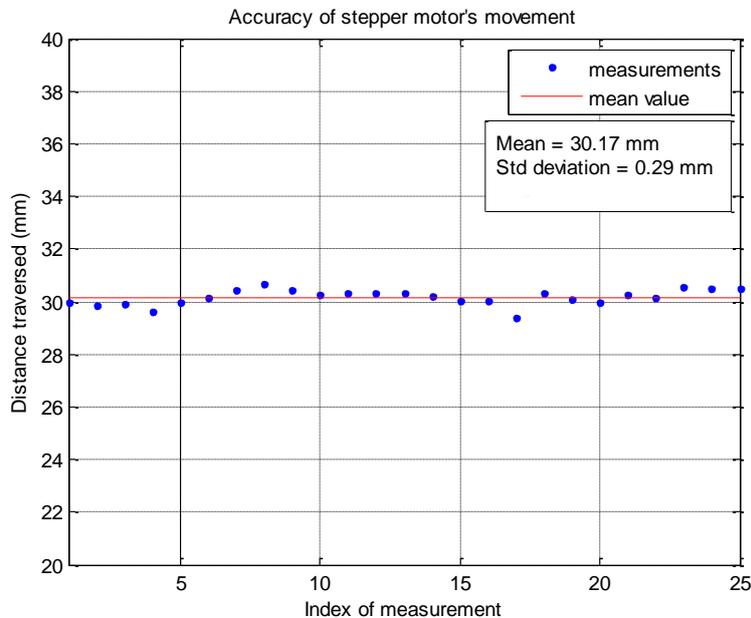


Figure 4.1.1 Accuracy of the bipolar stepper motor

In the figure above 25 measurements of the same movement/instruction are presented. For each measurement we instructed the motor to move 200 steps from the Reset position and then we

measured the distance it traversed. The mean value of the above distances is 30.17 mm which means that a single step of the motor is translated into approximately 0.15 mm. The standard deviation of the measurements is calculated at 0.29 mm, but this deviation is not caused only by the motor movement's inaccuracy but also from the error of measurement since they were taken manually<sup>1</sup>. According to the deviation calculated we can observe that the motor barely moves one step more/less every time it executes the same instruction. This kind of error cannot affect the function of the LVF since the slit that illuminates it has a 2 mm width. Which compared to the above deviation is 7 times larger. So the variance of the stepper motor's movement is accepted as accurate.

## 4.2 Output Light Power and System Throughput

In this paragraph we will study the light power that is produced by our system. Since we are using fiber optics and lenses to transfer the light through the device it is definite that there is a lot of light power scattered, dispersed and lost. In order to test this power we measured the light power in various points (parts) of the tunable light source, using a photometer.

Firstly, we measured the light at the output of the first fiber optic, meaning just before the light encounters the LVF. Then we measured the output of the source without the intervention of any filter. Lastly we measured the output light of the device intervening the LVF in various positions/wavelengths (from visible and 430nm to 790nm). The results of these measurements are presented here below.

Table 4.2.1 shows some of the values described above, as well as a percentage (throughput) according to the initial light power (light power of the fiber optic output before the LVF).

<b>Measurement</b>	<b>Watts</b>	<b>Percentage</b>
Fiber bundle before LVF	6.62	100%
Output fiber bundle w/o LVF	0.399	6.03%
Output with visible	0.016	0.24%
Max Output	0.0029	0.04%

**Table 4.2.1 Measurements of light power at various parts of the tunable light source**

Figure 4.2.1 shows the variation of the output light power in dependence with the center wavelength of the LVF. As it is shown the values of the output power are in the  $10^{-3}$  (milliwatt) order which is three orders of magnitude less than the initial light power (6.62 Watts  $\rightarrow$  2.90 mWatts). This decrease may seem as a big setback but as explained further below it is not.

---

<sup>1</sup> Although it is not very wise to take inaccurate measurements in order to test the accuracy of a system, we did not have a choice of setting and calibrating a stable and precise distance measuring mechanism.

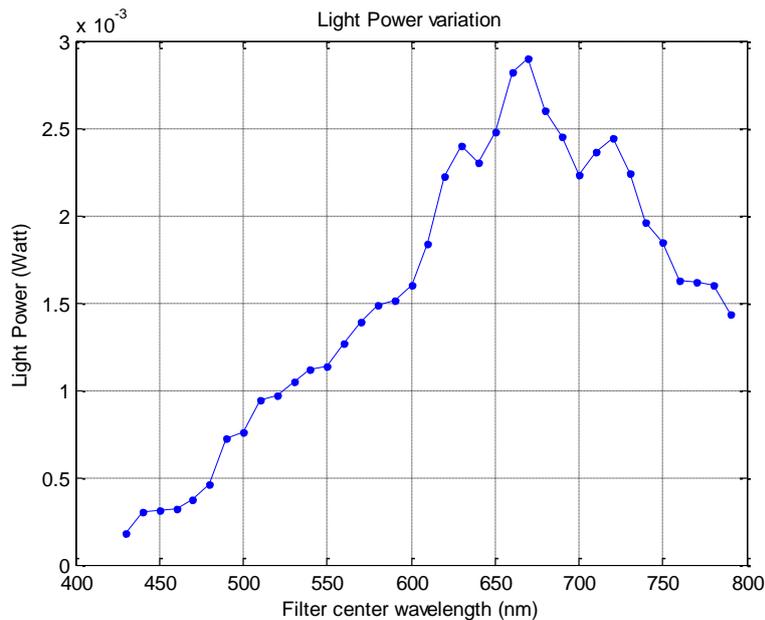


Figure 4.2.1 Output light power variation of the center wavelength of LVF

The power variation curve follows the form of the tungsten-halogen lamp transmittance curve (see Chapter 2.1). Indeed the lamp has one of its highest values of transmittance at around 660nm, where it's the highest point of this curve.

As we mentioned, there is a big decrease of the light power. This is because there is a lot of light dispersed during the transmission from the first fiber optic bundle to the cylindrical lens that gathers the light into a slit and on the LVF. Also during the according transmission on the opposite side, from slit to beam procedure, light is dispersed. Furthermore, it is accepted in an optical connection using fiber optics that there is a 50% loss of light power with every coupling, which explains a part of our big decrease of light power. Another factor that intuitively comes to mind, which causes loss of light power, is the transmittance of the LVF. When light is transmitted through the LVF then the output has a 40%-45% of its initial power. Although this is a point where power is lost, an - across the filter's length - transmittance of 40% or 45% is considered high throughput. Concluding, the decrease of light power is primarily caused by the optical connections between the components of the tunable light source. The loss in optical connections could be optimized to the minimum for a greater system throughput, but this is something that was not attempted in this diploma thesis.

In terms of output light power the system throughput is not high, but the resulting output light power is proven to be more than enough for microscope illumination purposes and spectral acquisition. Any improvement on the light power loss would increase the efficiency of our device but even with this kind of loss, the tunable light source we constructed, is very efficient in terms of spectral microscopy. The important observation is that this efficiency is achieved only by using a 150W 21V halogen lamp light source and a commonly used (relatively cheap) spectral camera.

### 4.3 System Frame Rate

The last evaluation issue we studied is presented in this paragraph. When running the system in spectral cube acquiring mode (scanning across the length of the LVF), it is essential to know at which frame rate our device is registering/saving the spectral images of the scan. The system's frame rate is dependent to the camera frame rate; the quicker images are sent from the camera to the PC, the quicker the images are saved. It is desired that the device can scan several different bands almost simultaneously, so a high frame rate is wanted.

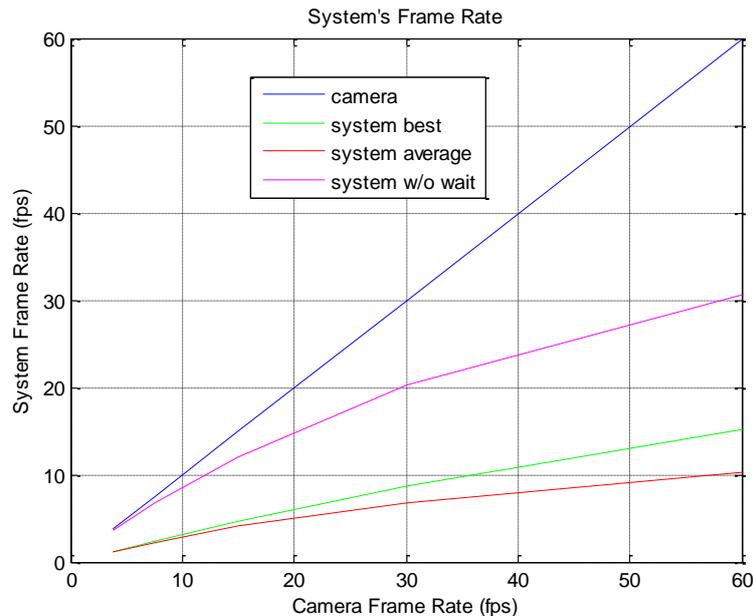


Figure 4.3.1 Frame Rate of our spectral imaging system

The figure above presents the variation of our system's frame rate over the frame rate of the camera, which varies from 3.75 to 60 fps. The blue line shown above is the ideal case, where the spectral imaging system could save spectral images as fast as the camera sends them. It is quite obvious that with our hardware configuration we could not achieve something like that performance, since in any case the filter movement will insert a minimum amount of delay. The green and the red data are the best and the average case respectively of our system's frame rate. As a best case we consider a scan with very small wavelength step, therefore a small amount of delay inserted by the motor. As average case we consider a scan with 20nm wavelength step. More specifically the best case (2nm) inserts 16 ms delay for every transition from one step to the next and the average case (20nm) inserts 47 ms. These delays would not cause our system to slow down so much as shown in Figure 4.3.1, but as we mentioned in Chapter 3 the software waits three frames before the desired image is saved. The necessity of these waiting frames was discovered upon testing, and it seems that the spectral camera holds frames in its internal memory before sending to the PC with a delay of three frames. Lastly, the magenta line shows how our system would respond (in a best case scan) if there were no waiting frames of the camera and the only delay inserted would be the one of the motor. The figure above is summarized in the next table.

<b>Case</b>	<b>Camera FPS</b>	<b>System FPS</b>
Average case	7.5	2.23 (29.73%)
Best case	7.5	2.40 (32.00%)
W/o waiting 3 frames	7.5	6.70 (89.33%)
<hr/>		
Average case	60	10.31 (17.18%)
Best case	60	15.15 (25.25%)
W/o waiting 3 frames	60	30.61 (51.01%)

Table 4.3.1 Summary of Figure 4.3.1

## Chapter 5

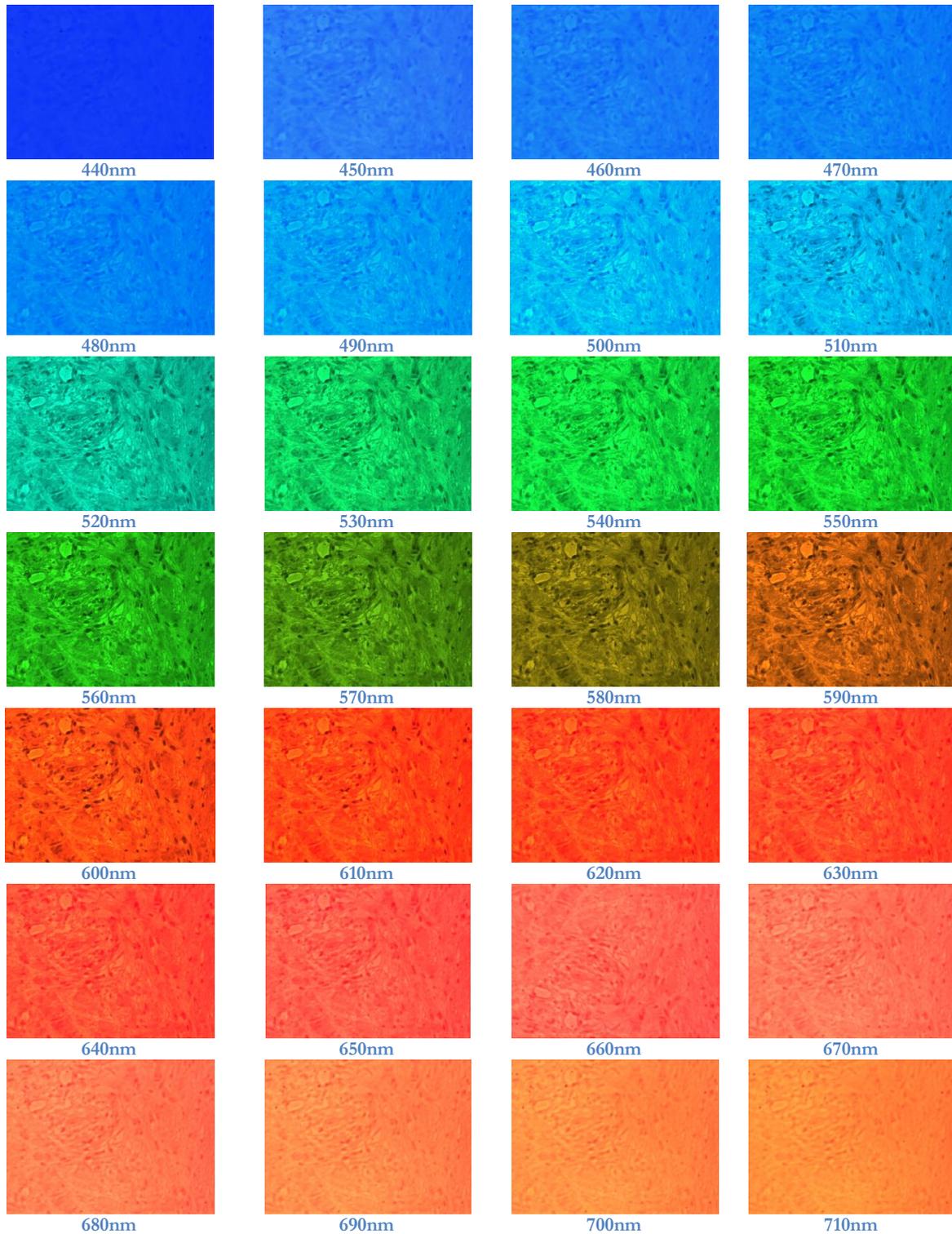
### Application in Medical Histology

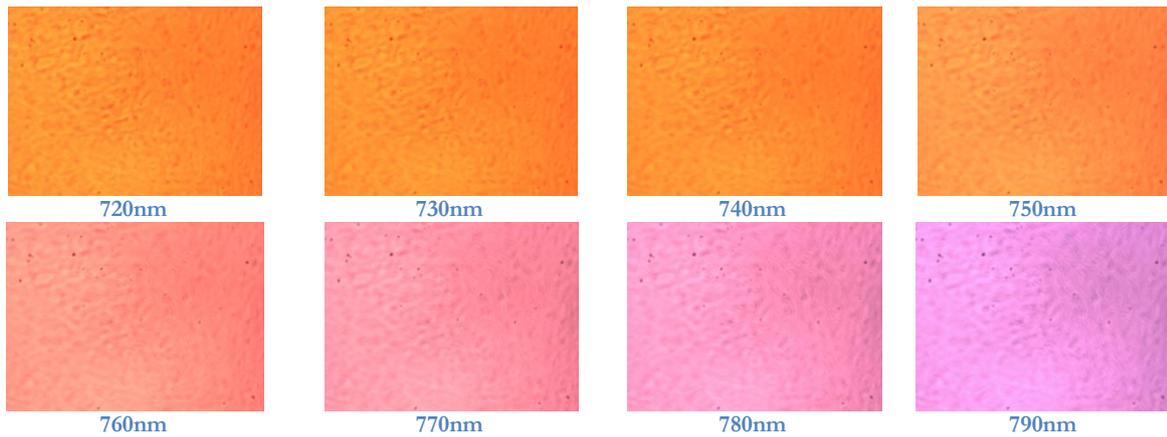
As we aforementioned spectral imaging finds great application in bio-medicine regions such as medical histology. The system presented in this thesis is applicable to many kinds of bio-medicine issues as well as histology. This chapter describes the measurements taken and the methods followed in order to prove this concept. In paragraph 5.1 various color images of the specimen of the microscope excited in different wavelengths are demonstrated. Paragraph 5.2 presents the spectral imaging capabilities of our system using the spectral camera described in paragraph 2.5. The spectral data acquired are processed and diagnostic pseudo-color maps are produced. Lastly, paragraph 5.3 presents the excitation/emission spectroscopic capabilities of our system using as imaging device the hyper-spectral imager MuSIS HS. Also in 5.3 diagnostic pseudo-color maps are presented.

In the following sections various images of microscope's specimens are presented. All the following images are regions of the same microscope slide on which there is disposed sample of breast cancer tissue [\[LINK\]](#). This sample has been processed with an immunohistochemistry (IHC) procedure in order to highlight the cancer and the non-cancer cells. IHC refers to the processes that detect antibodies (e.g. proteins) in cells of a tissue section by exploiting the principle of antibodies binding specifically to antigens in biological tissues [\[LINK\]](#). An antibody is a glycoprotein used by the immune system to identify and neutralize substances foreign to the body, such as bacteria and viruses and other infectious agents, known as antigens. In order for the antigen-antibody immunoreaction to be seen in a microscope the antibody must take a label – enzyme for light microscopy. An enzyme label can be visualized in the light microscope by means of enzyme histochemical methods via chromogenic reactions. In enzyme-histochemical reactions, a soluble colorless substrate is converted into a water-insoluble colored compound either directly or in coupled reaction [\[LINK\]](#). Histochemical detection of estrogen receptors [\[LINK\]](#) (which are the desired characteristic to be highlighted [\[LINK\]](#)) is based on the conversion of the amine; diaminobenzidine (DAB) into water-insoluble pigments in the presence of hydrogen peroxide ( $H_2O_2$ ) [\[LINK\]](#). The resulted color of DAB is red/brown. The anti-chromophore substance is hematoxylin Mayer which is used to contrast the non-cancer cells.

## 5.1 Sample Images of the Microscope Configuration

In this section we will demonstrate some sample result/images of the TLS-microscope-spectral imager configuration. These results demonstrate the range and tunability of the system constructed.





The above images are just a sample of the system results. In the next two paragraphs results with diagnostic value are presented and compared.

## 5.2 Spectroscopic Results

In this section spectral data and diagnostic pseudo-color map production will be presented. All the images in this paragraph were acquired using the simple spectral imager integrated in our system and described in paragraph 2.5. These measurements are taken in order to confirm the spectral and diagnostic capabilities of the configuration designed and developed as part of this thesis. Firstly, a zoom-in of an RGB image (color image, filter wavelength → Visible) of a region of a microscope slide (as described in introduction of Chapter 5) is presented in Figure 5.2.1. Also the figure below presents the spectra of each class we need to discriminate in order to produce the diagnostic pseudo-color map.

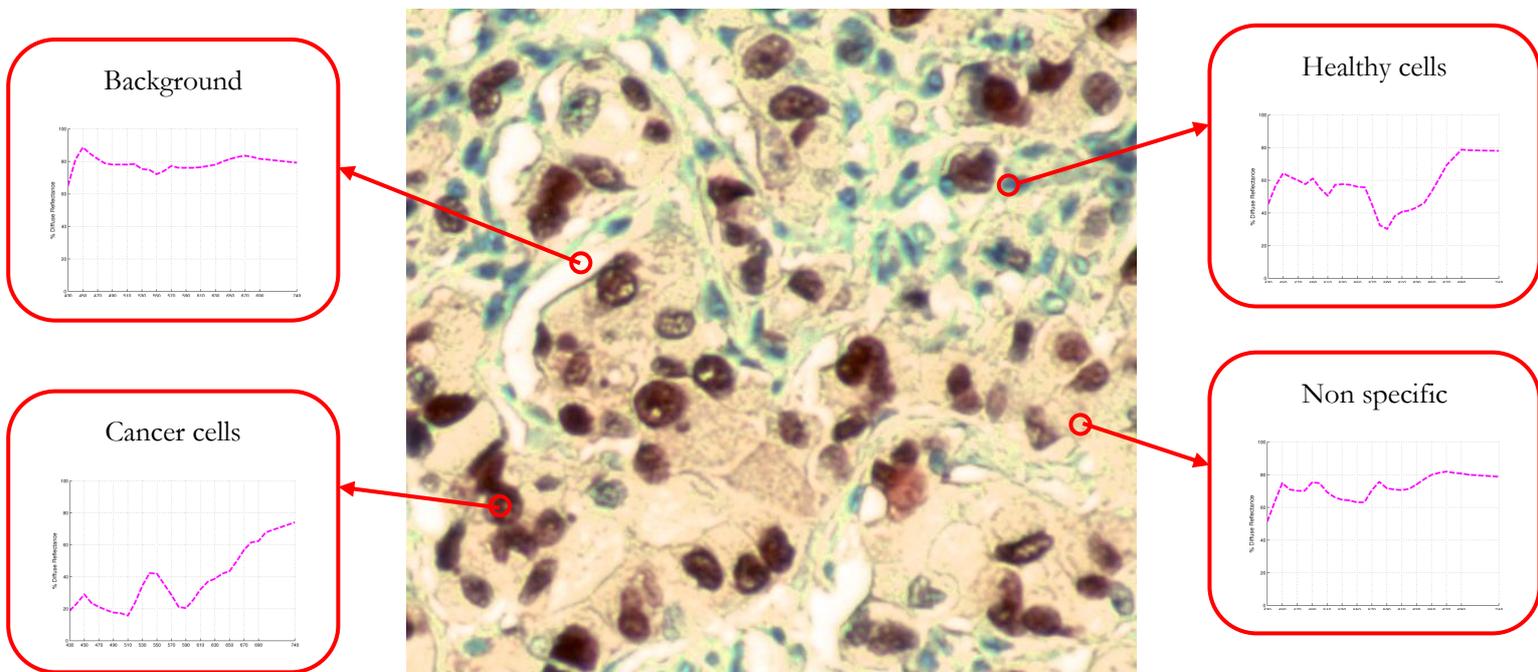


Figure 5.2.1 RGB image of the region of the slide tested

The spectra presented above and the diagnostic maps that will be presented furthermore are produced using the ELGRECO tool introduced in paragraph 3.5. This tool is used provides the capability of defining classes of spectra in a spectral image in order to classify items according to their spectra. Most of the maps presented here on have four classes (cancer cells, healthy cells, non specific material and background). After acquiring the above color image, a spectral cube was acquired by executing a scan across the length of the LVF and registering a spectral image every 10nm. Some of these spectral images are presented here below. The spectral cube is not presented in its whole because of ease of presentation purposes.

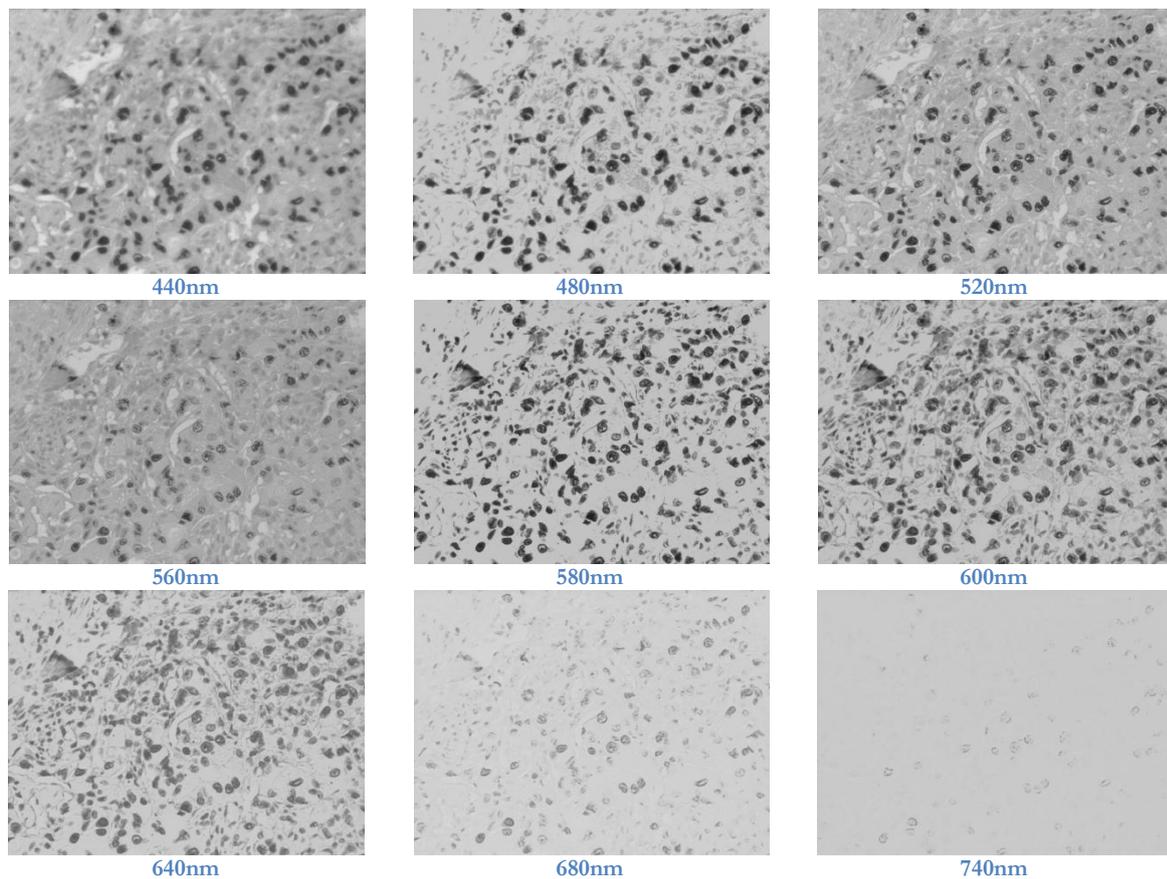


Figure 5.2.2 Spectral images with various excitation wavelengths

The complete (normal zoom) of the RGB image of Figure 5.2.1, the spectra figures of the area/classes of the microscope slide and all the images of the above spectral cubes are cited in Appendix A.

The acquired spectral cube is given as input to the ELGRECO tool and then two kinds of maps are produced. The first (Figure 5.2.3) is result of an unsupervised classification algorithm; k-means. This algorithm tries with an exhaustive method to classify the features of a spectral cube. The second (Figure 5.2.4) is result of supervised classification algorithm; Spectral Angle Mapper (SAM) algorithm [LINK2]. SAM needs a training set of spectra of every class, as an additional input. In Appendix A there are pseudo-color maps produced using other classification algorithms.

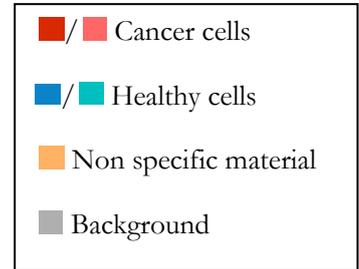
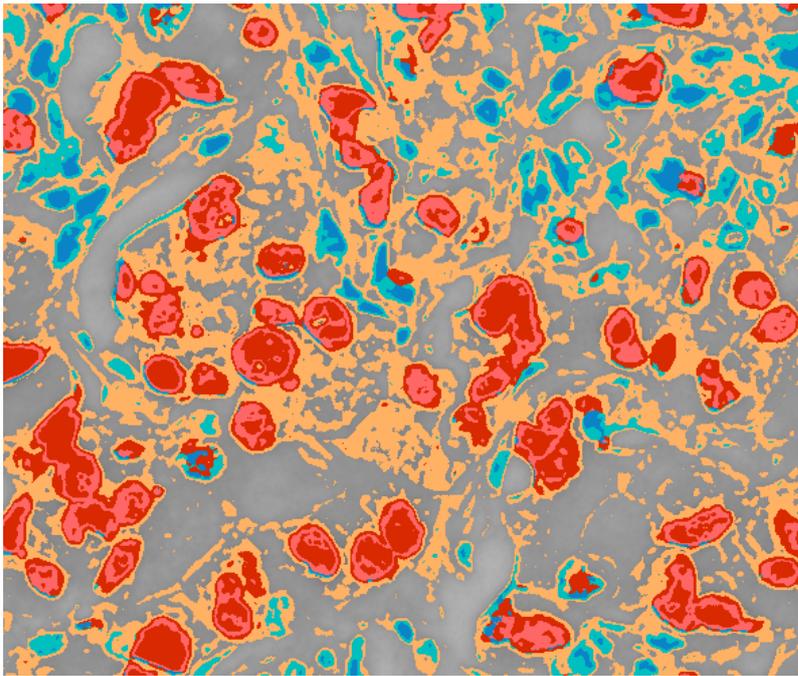


Figure 5.2.3 Diagnostic pseudo-color map produced by a spectral cube obtained with the TLS-DragonFly configuration and classified using k-means algorithm

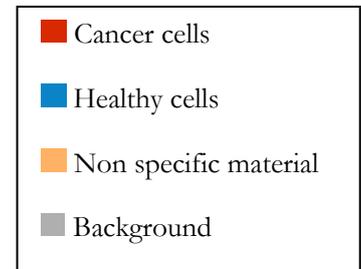
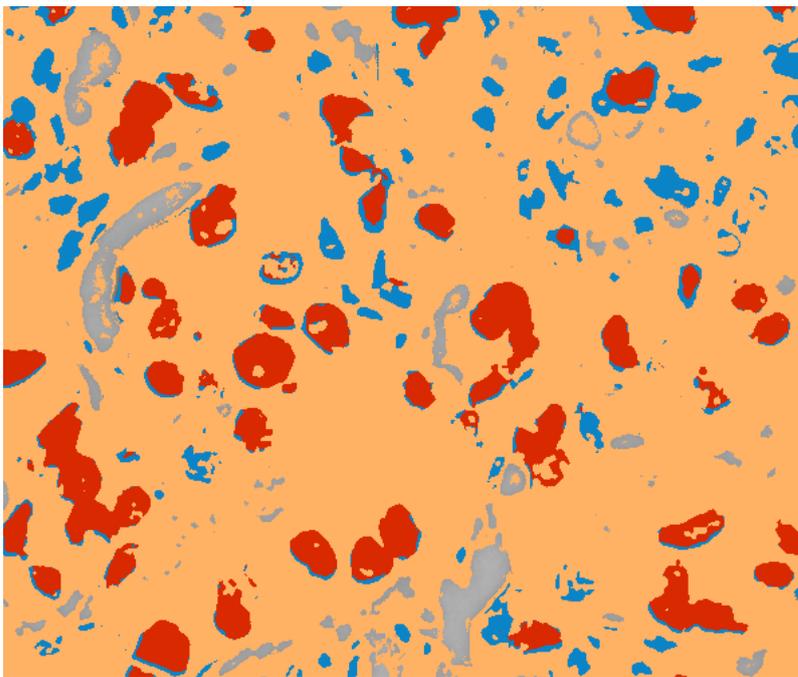


Figure 5.2.4 Diagnostic pseudo-color map produced by a spectral cube obtained with the TLS-DragonFly configuration and classified using SAM algorithm

In figures above the map is zoomed-in on the same area as the one presented in the RGB image of Figure 5.2.1. The extended diagnostic maps are presented in Appendix A.

### 5.3 Excitation/Emission Microscopy Results

In this paragraph an excitation/emission configuration is presented, which incorporates our TLS design. The application of the TLS system in an excitation/emission microscopy is simply done by changing the imaging device (DragonFly originally) with a hyper-(or multi-)spectral imaging device such as MuSIS HS. This special imaging device has the capability of filtering (with various mechanically tunable filters) the light before interacting with the CCD sensor. With the use of MuSIS HS the configuration has the capability to tune the excitation and scan the emission (light transmitted through the microscope's specimen) wavelength.

MuSIS HS is a powerful – in terms of spectral range – imaging and acquiring tool. More specifically several key specifications of this spectral camera are presented here below.

Property	Value / Description
Spectral Range	360nm – 1550nm
FWHM	5nm
Resolution	1600x1200 real-time capturing
Transmittance	≥40%
Camera	Megapixel CCD camera with IEEE-1394 data transfer
Calibration	A special calibration procedure <a href="#">[LINK]</a> is executed before any imaging procedures, compensating for the wavelength dependence of the response of the electro optical parts of the system

Table 5.3.1 MuSIS HS specifications

The specimen which is studied in the following images and spectral cubes presented is the same one that is studied in the previous paragraph. So firstly a zoom-in of an RGB image (color image, filter wavelength → Visible) of a region of the slide is presented in the next figure.

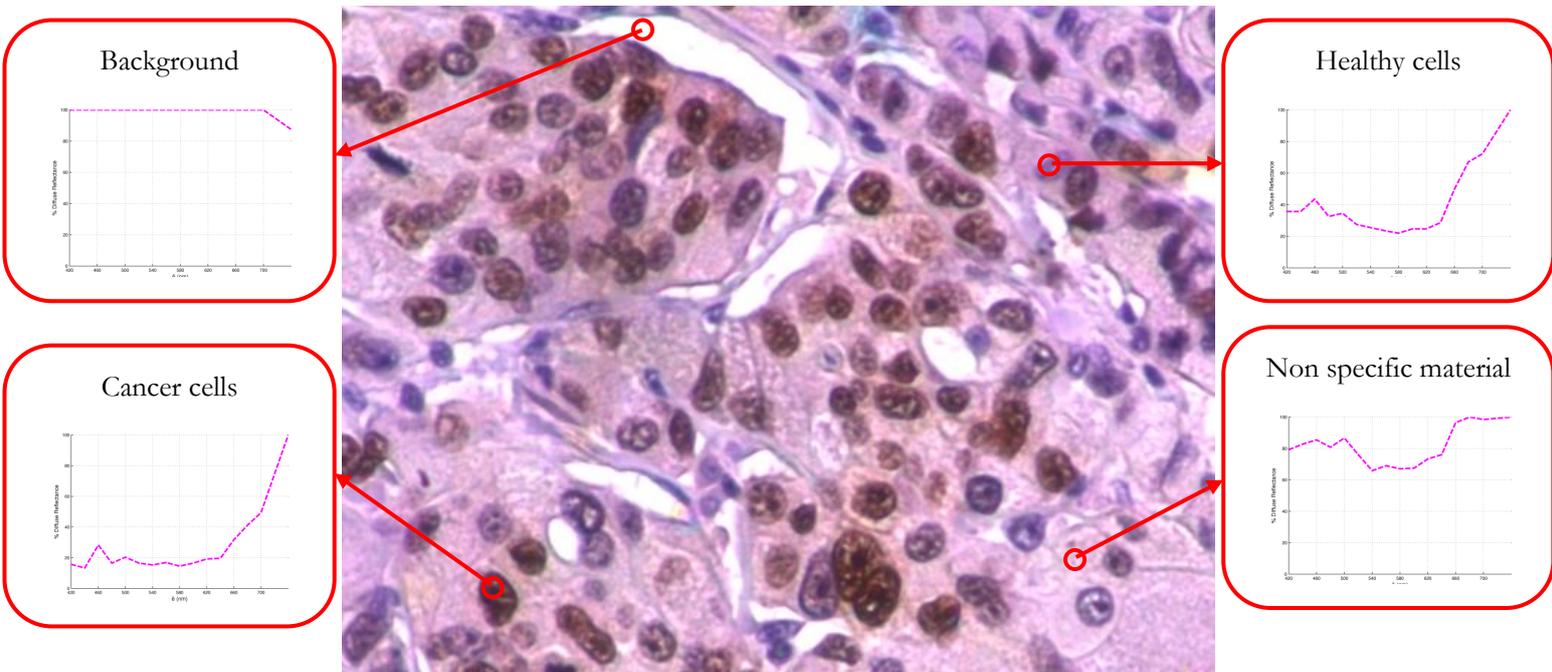


Figure 5.3.1 RGB image of the region of the slide tested with MuSIS HS device

Part of the according spectral cube of the slide is presented below. The complete cube and the complete (normal zoom) RGB image of Figure 5.3.1 are cited in Appendix A.

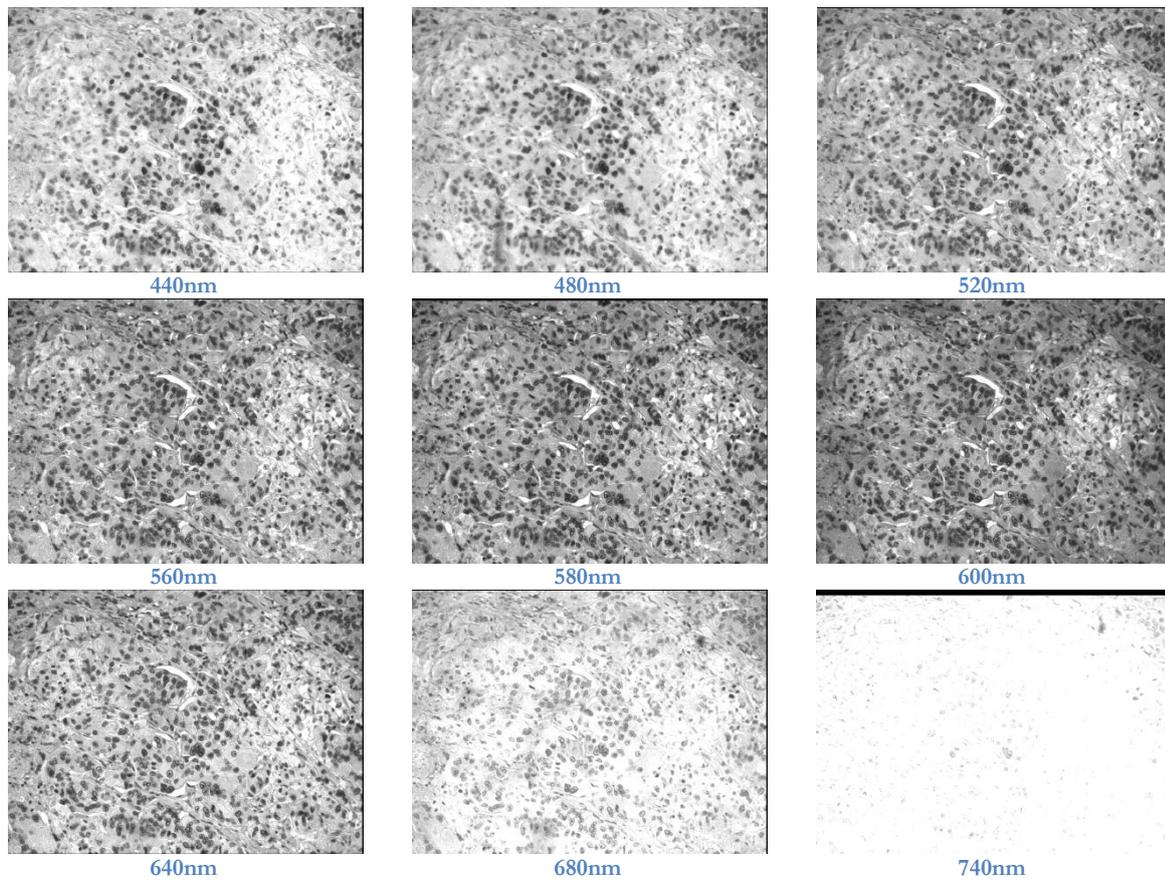


Figure 5.3.2 Spectral images with various wavelengths acquired with MuSIS HS

Following the same procedure as in paragraph 5.2, two diagnostic pseudo-color maps (k-means and SAM) are presented. These following maps are basically a result with tested diagnostic value [\[LINK\]](#) and they are presented in order to compare them with the results of our configuration (TLS- DragonFly and TLS-MuSIS HS).

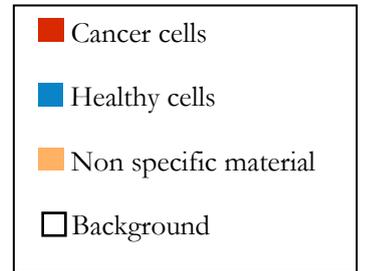
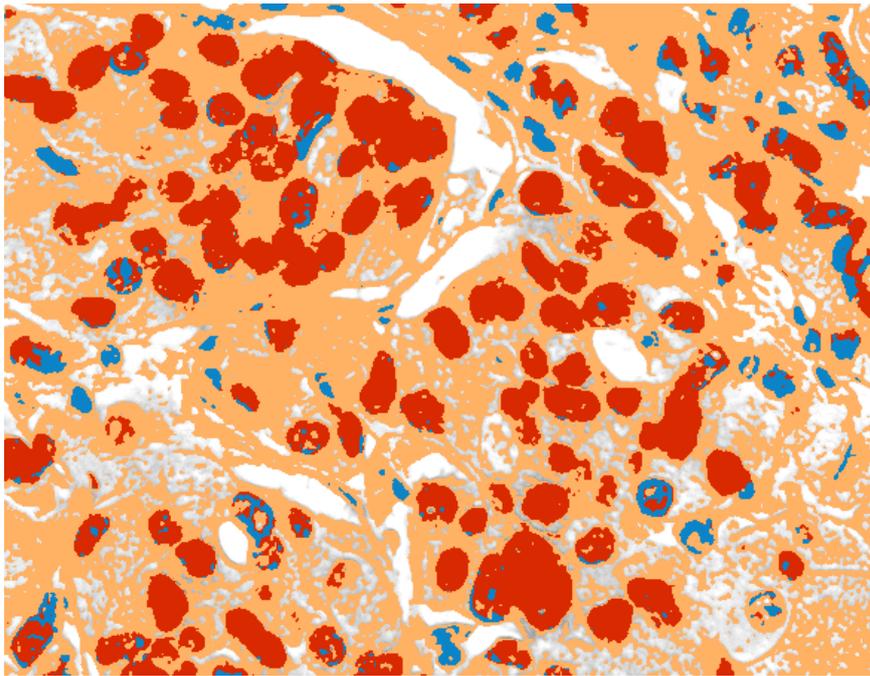


Figure 5.3.3 Diagnostic pseudo-color map produced by a spectral cube obtained with the MuSIS HS device and classified using k-means algorithm

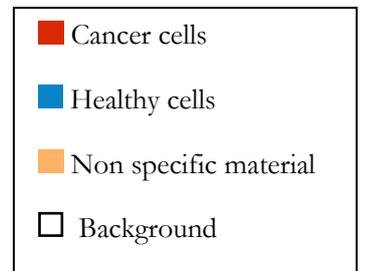
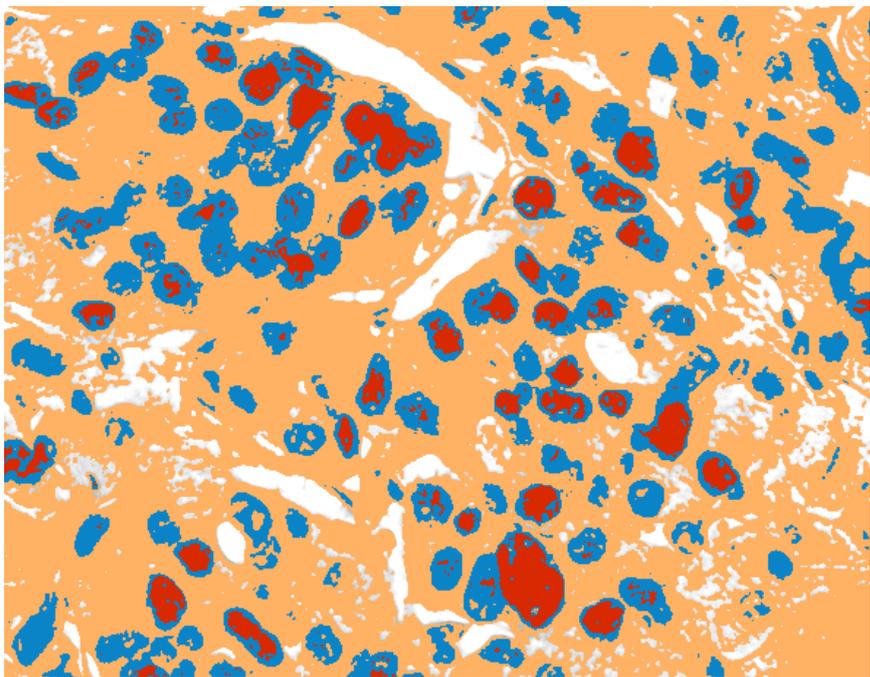


Figure 5.3.3 Diagnostic pseudo-color map produced by a spectral cube obtained with the MuSIS HS device and classified using SAM algorithm

In figures above the map is zoomed-in on the same area as the one presented in the RGB image of Figure 5.3.1. The extended diagnostic maps are presented in Appendix A.

In order to prove the concept of application of our TLS configuration in excitation/emission microscopy we acquired several spectral cubes using the MuSIS HS with different excitation wavelengths by tuning the center wavelength of the LVF. In this section three spectral cubes are presented, each acquired with excitation wavelengths 430nm, 570nm, and 700nm accordingly. All three cubes were acquired on the same region of the slide shown in Figure 5.3.1. As it is expected only few bands of each spectral cube has valuable spectral data, since the microscope's specimen does not emit in any other bands (does not contain any fluorescent substances) than the area around the excited wavelength. The spectra of the cancer cells for each spectral cube are presented in the next figure.

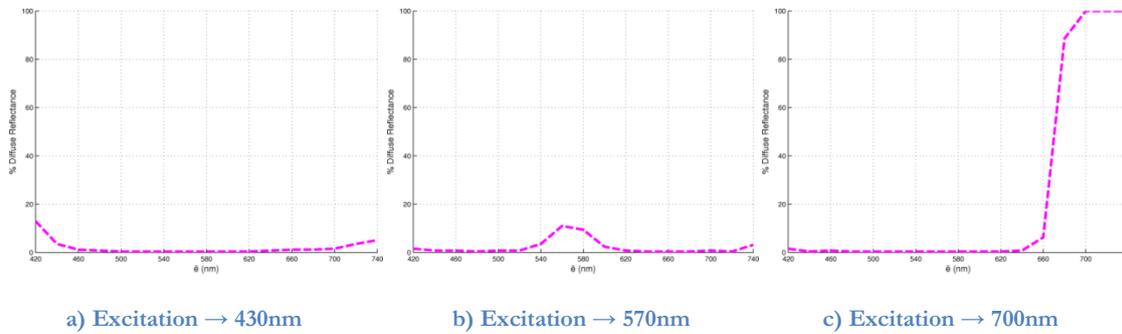


Figure 5.3.4 Cancer cell spectra in the three spectral cubes acquired with the TLS-MuSIS HS configuration

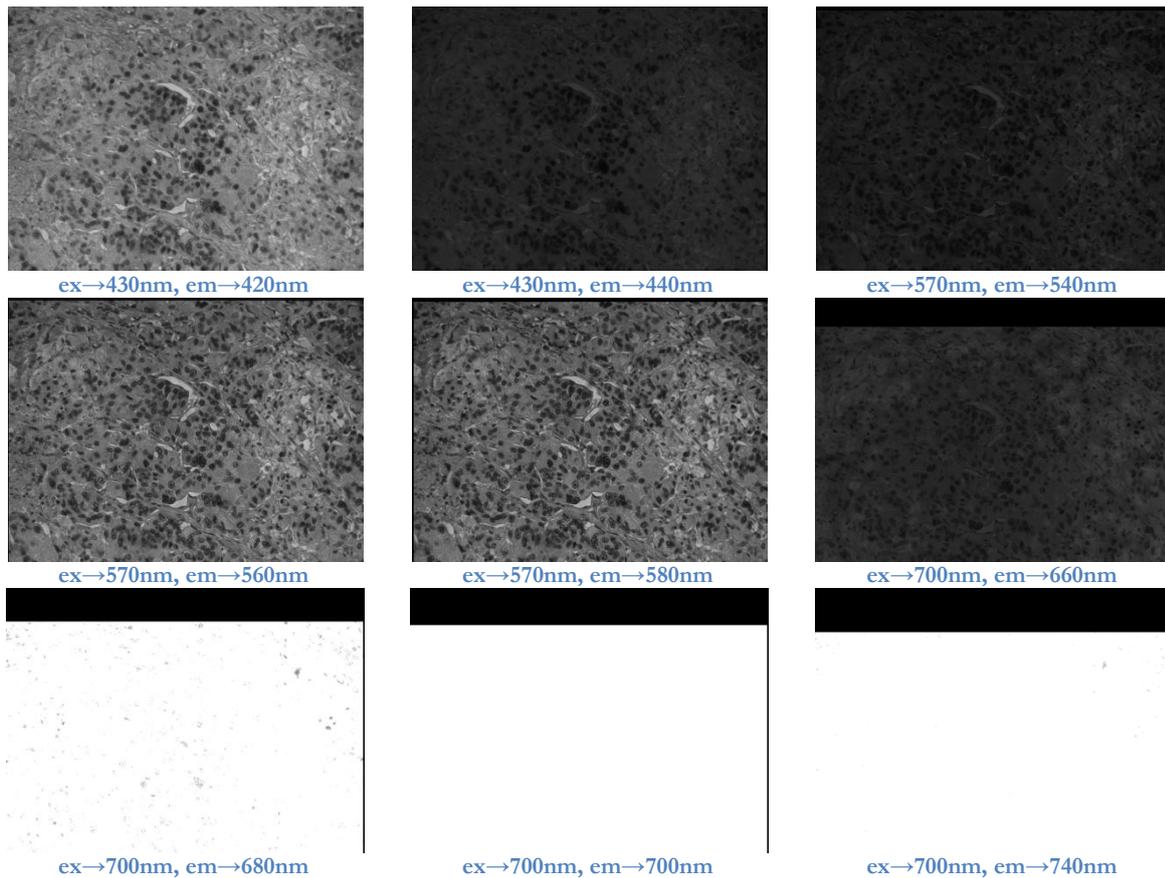


Figure 5.3.5 Spectral images with various excitation and emission wavelengths acquired with TLS-MuSIS HS configuration

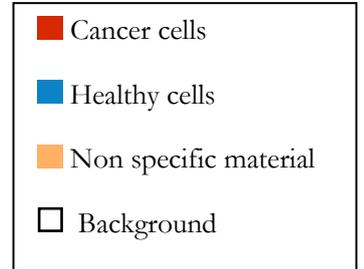
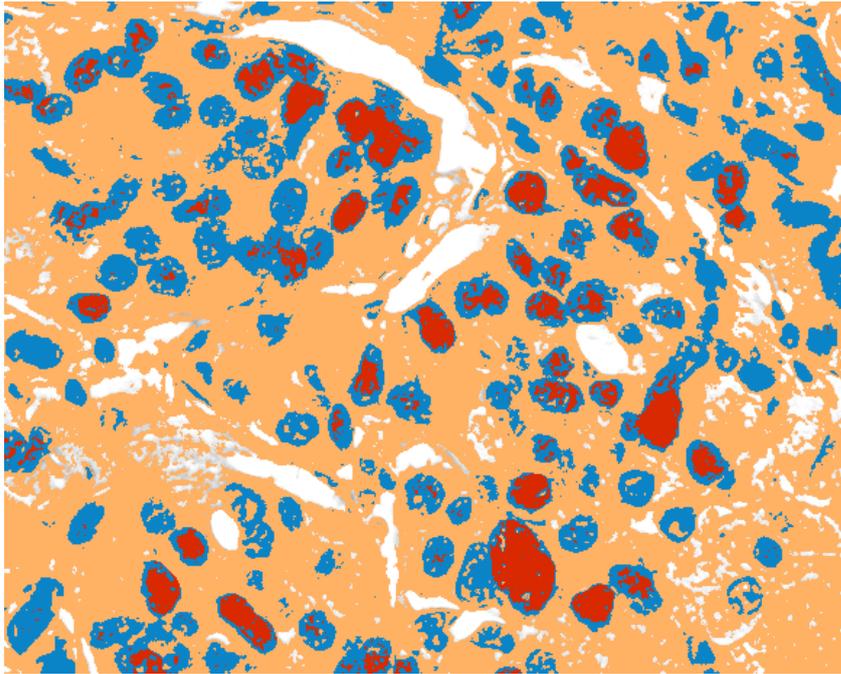


Figure 5.3.6 Diagnostic pseudo-color map produced by selected bands of three spectral cubes obtained with the TLS-MuSIS HS configuration and classified using SAM algorithm

In the figure above the Spectral Angle Mapper classification algorithm is executed with three selected spectral images from the three different cubes (420nm, 580nm and 700nm). The map presented in Figure 5.3.6 has almost the same diagnostic quality of the maps produced by the MuSIS HS device. This proves that our system provides the capability of scanning in various wavelengths of excitation and if combined with a spectral imaging device as MuSIS HS those scans can provide useful diagnostic information. More specifically using a TLS-MuSIS HS with non-fluorescent specimens enables the researcher to define which bands are the most important and contain the most crucial – in terms of detection - spectral information. If the TLS-MuSIS HS is used with fluorescent specimens then fluorescence microscopy is enabled with a great range of both excitation and emission wavelength selection.

## Chapter 6

### Conclusions and Future Work

In this diploma thesis a tunable light source and its integration with a spectral imager and a microscope are presented. The system is built on a key component; a linear variable band filter as monochromator of the light source. Controlling a stepper motor, this filter is moved resulting to the shifting of the center wavelength of the filter and thus the change of the output light's wavelength. This tunable source is integrated into a system comprising a commonly used microscope and a spectral camera. The system is tested for use in spectroscopic microscopy and also in excitation/emission (fluorescence) microscopy.

Our configuration is a cost-effective approach in hardware and software demanding areas of biomedicine, such as fluorescence microscopy. The device presented has accurate tunability over a wide range of spectral bands, high output light, high quality of spectral data and fast acquisition rates. It is evident that the characteristics presented above have a trade-off relation, but the balance of our configuration among these parameters seems to be very efficient for the purposes used.

We use a common CCD spectral camera unlike the configurations which use AOTF, which incorporate an EM-CCD camera which is much more expensive. Also the device can be tuned in a wide spectral range (400-1000nm) with an adequate frame rate, unlike other configurations that can switch excitation bands very quickly (femtosecond) but only have a range of two or three spectral bands. Lastly our system provides adequate illumination for microscopy purposes along with clear spectral data because of the simple optical path that the light follows before it is transmitted through the specimen (this way there is no noise inserted in the light signal).

In terms of microscopy configuration, the developed TLS proved to have similar results with ones yielded from MuSIS HS which is a powerful spectral imaging tool. Also the combination of TLS and MuSIS HS proved to have great possibilities in terms of acquiring excitation/emission spectral data.

As future work, there are a lot of improvements to made hardware and software-wise. The light power throughput of the tunable light source can be significantly improved by carefully re-designing the output fiber optic bundles that lead the light on the filter. Also an improvement on the power throughput would be the replacement of the halogen lamp source with a super-continuum laser source which produces a broadband light (Figure 6.1) with some very useful characteristics. A super-continuum laser would be the ideal light source for our configuration since its output is directional and coherent. This way the tunability of the TLS will also improve as well as the light dispersions will be reduced. Probably the last important improvement in terms of hardware would be the full integration of the TLS with a device like MuSIS HS so the excitation and emission filtering can be simultaneously tuned. With this integration a wide range fluorescence microscopy would be enabled.

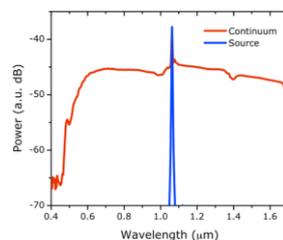


Figure 6.1 Super-continuum laser's spectrum

Furthermore, the automated calibration operation should be revised, not only to fix some small bugs, but also to reconsider the algorithmic part of this operation. The algorithm is presented in this thesis is adequate but can surely be improved. One improvement would be to gather information for the image reflectance from a broader and more random area than a few central pixels. Also the algorithm could be speeded up by a smarter searching and selection of the camera settings each time. This could be done by taken in account the previous selected values. Lastly when calibrating and in order to acquire really clear spectral data it is essential to correct the lens transmittance which decreases the reflectance as we move away from the center of the image.

## Appendix A

This appendix presents several more results from TLS configurations and the MuSIS HS device.

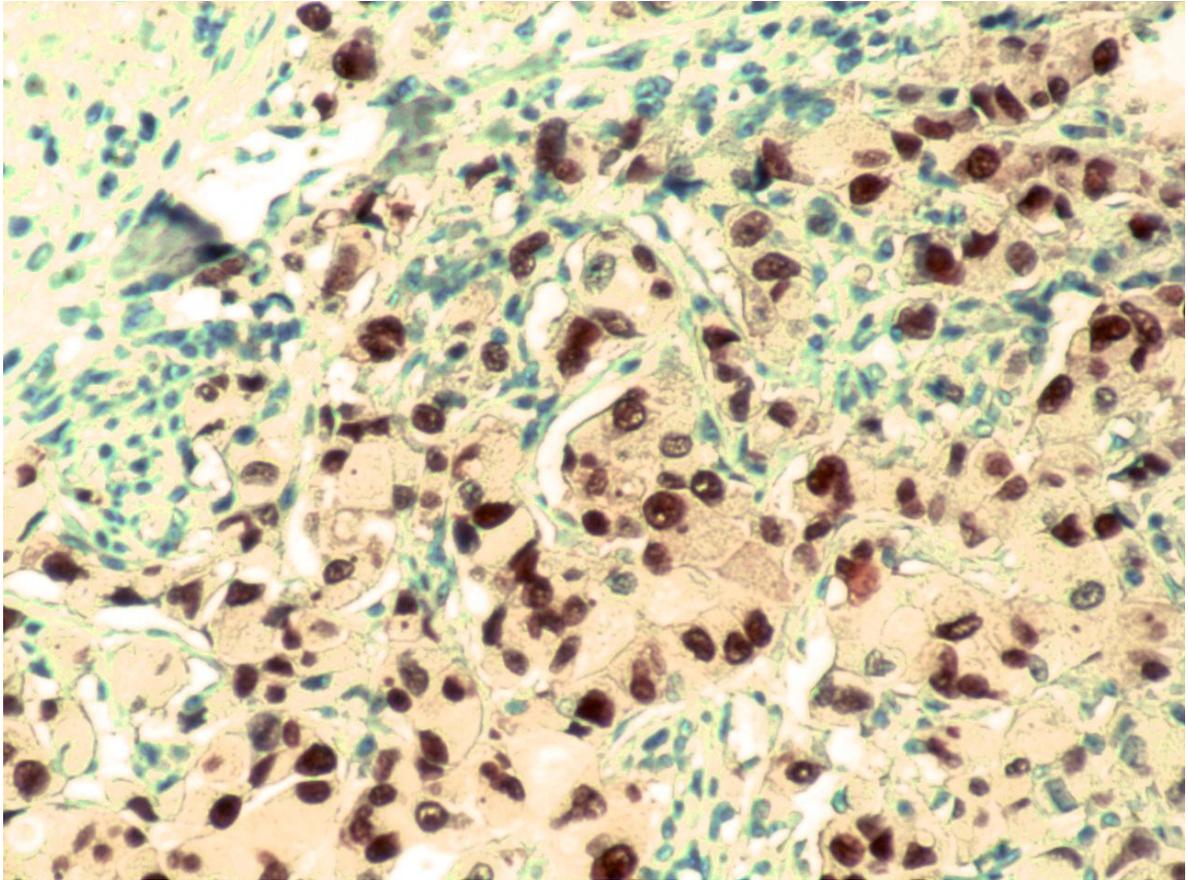


Figure A.1 Complete (normal zoom) shown in Figure 5.2.1

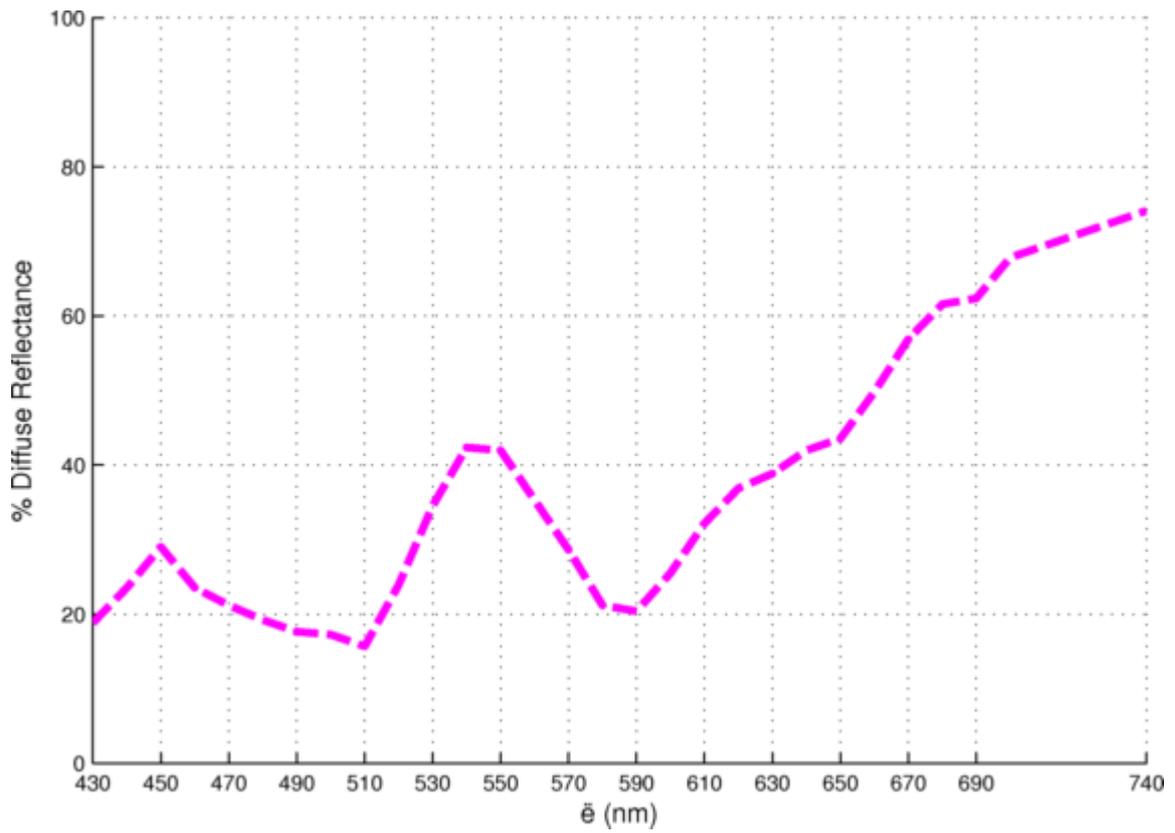


Figure A.2 Spectrum of cancer cells in Figure 5.2.1

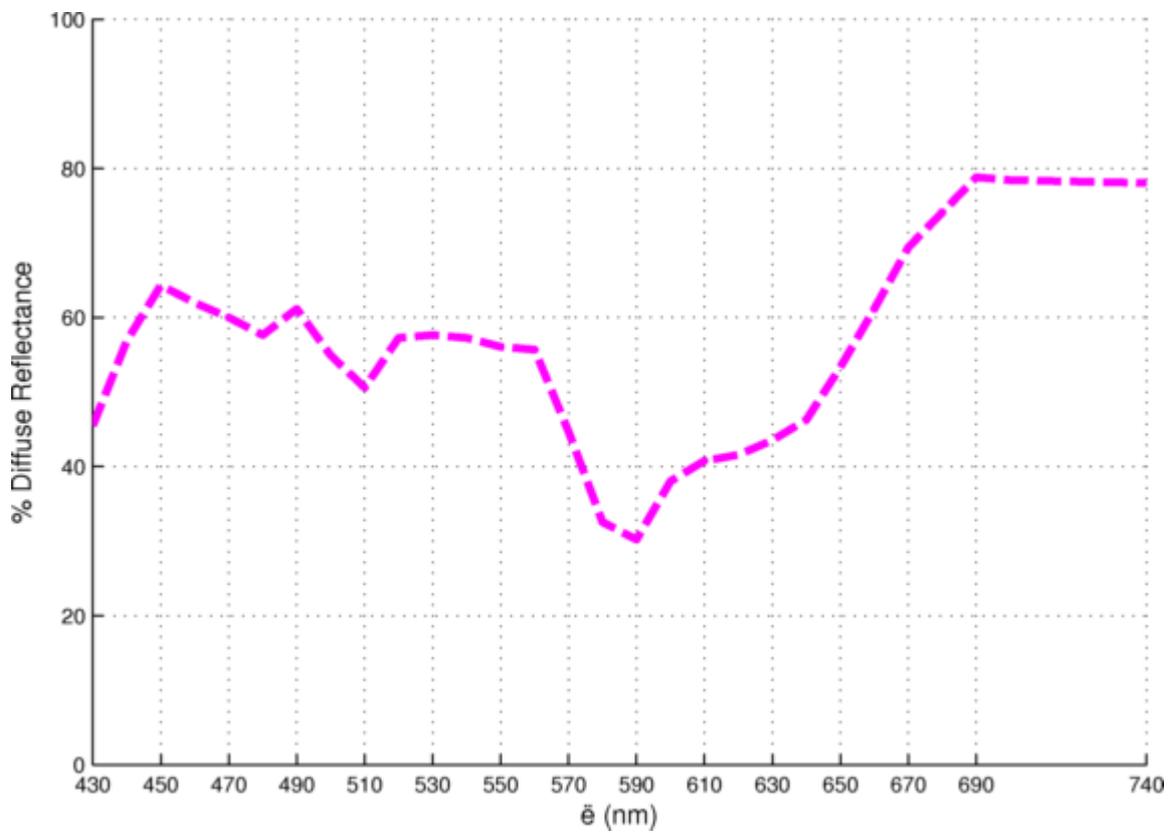


Figure A.3 Spectrum of health cells in Figure 5.2.1

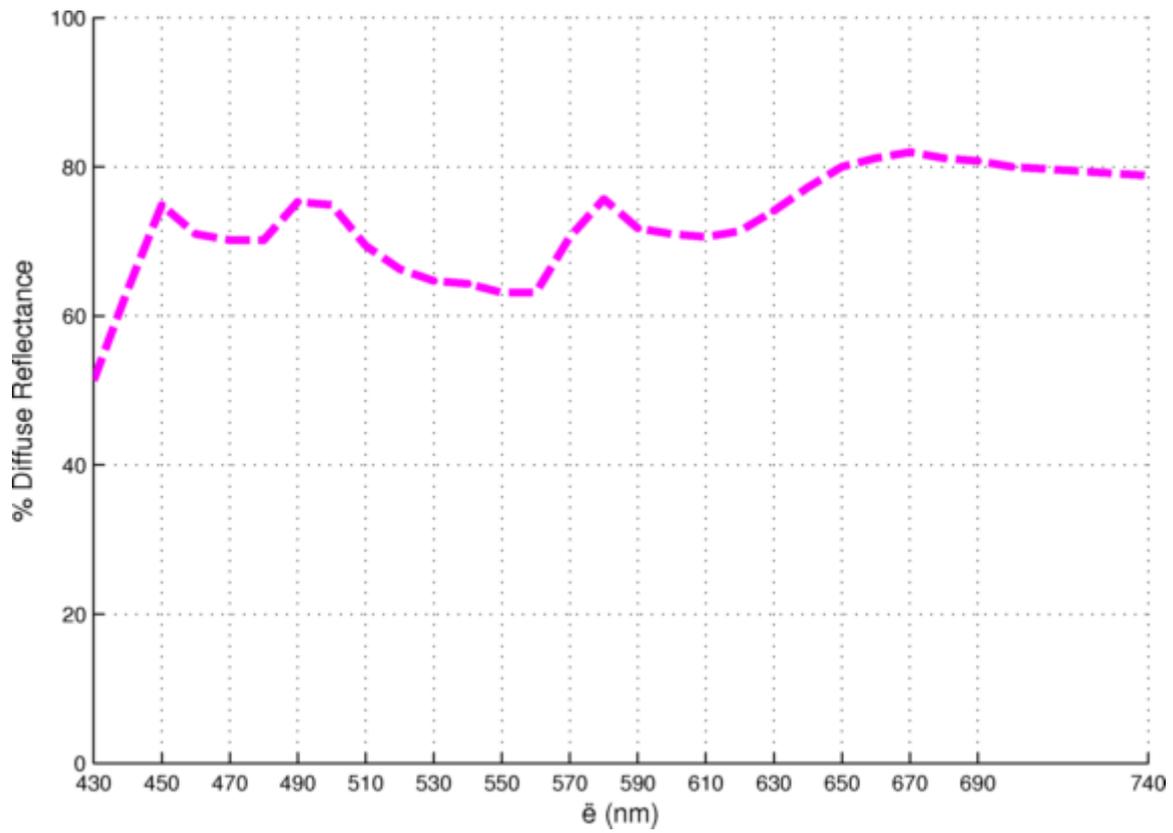


Figure A.4 Spectrum of non specific material in Figure 5.2.1

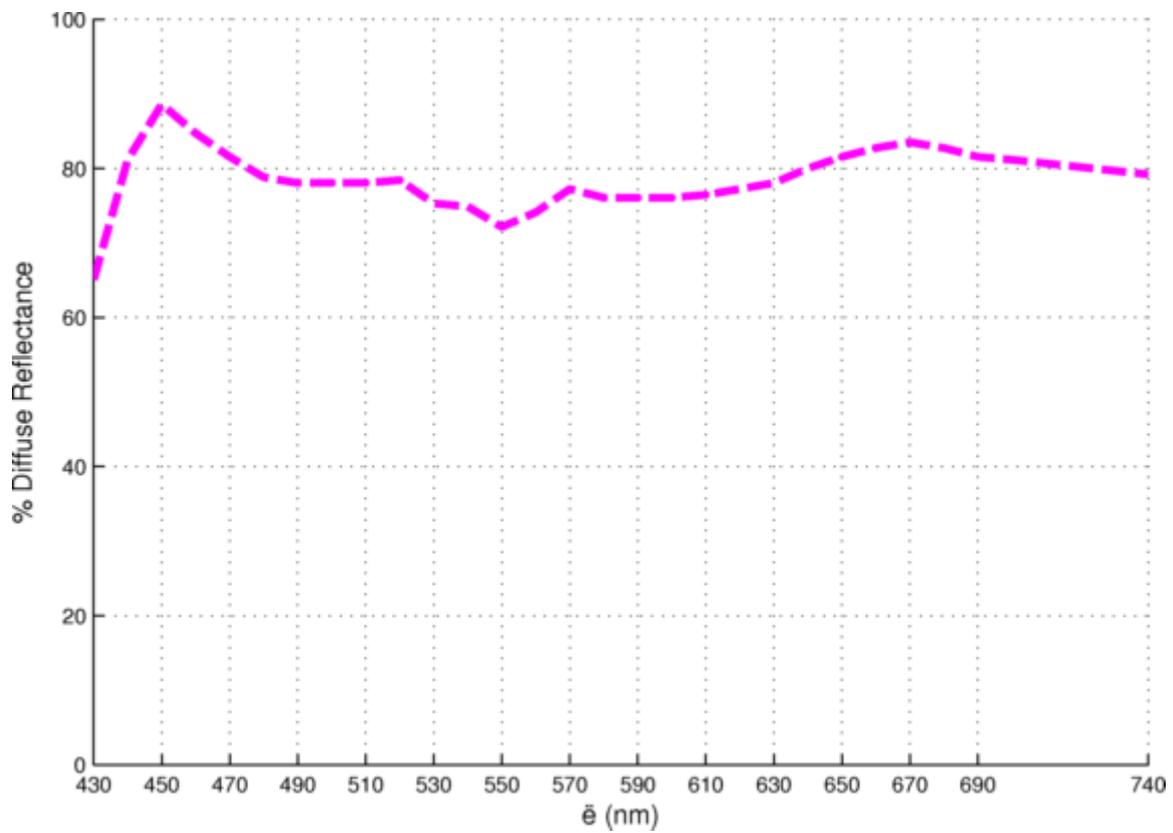
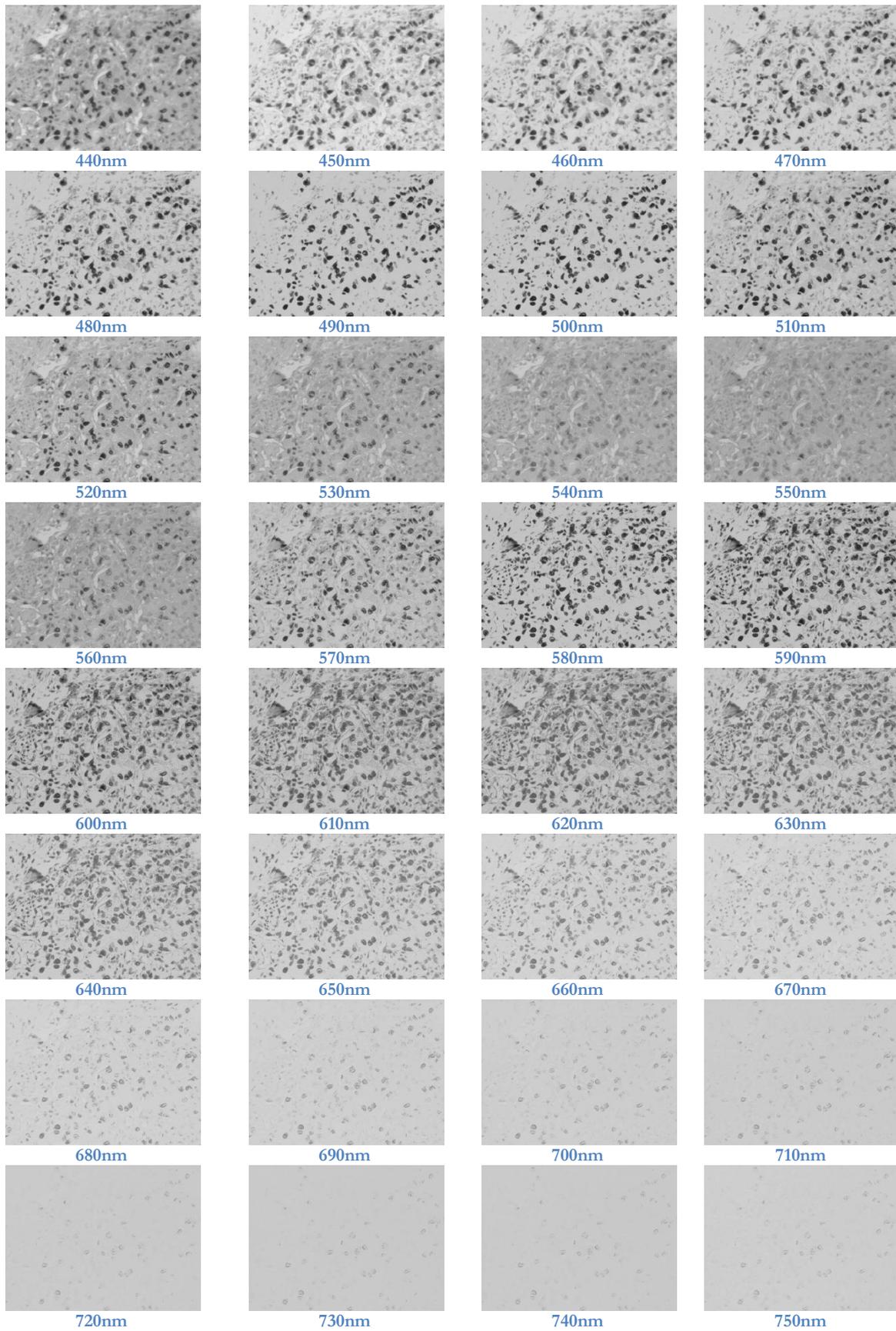


Figure A.5 Spectrum of background in Figure 5.2.1



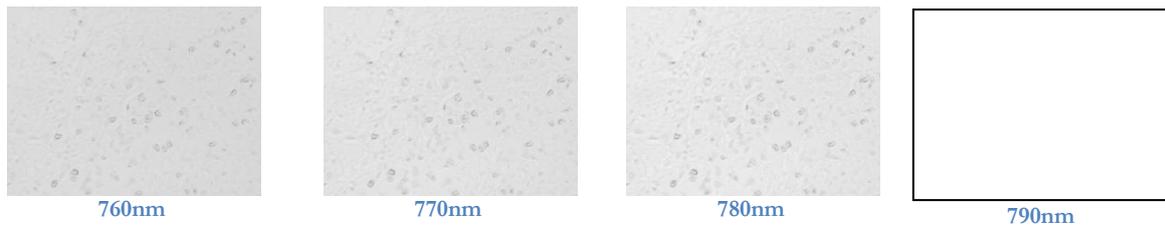


Figure A.6 Complete spectral cube presented in Figure 5.2.2

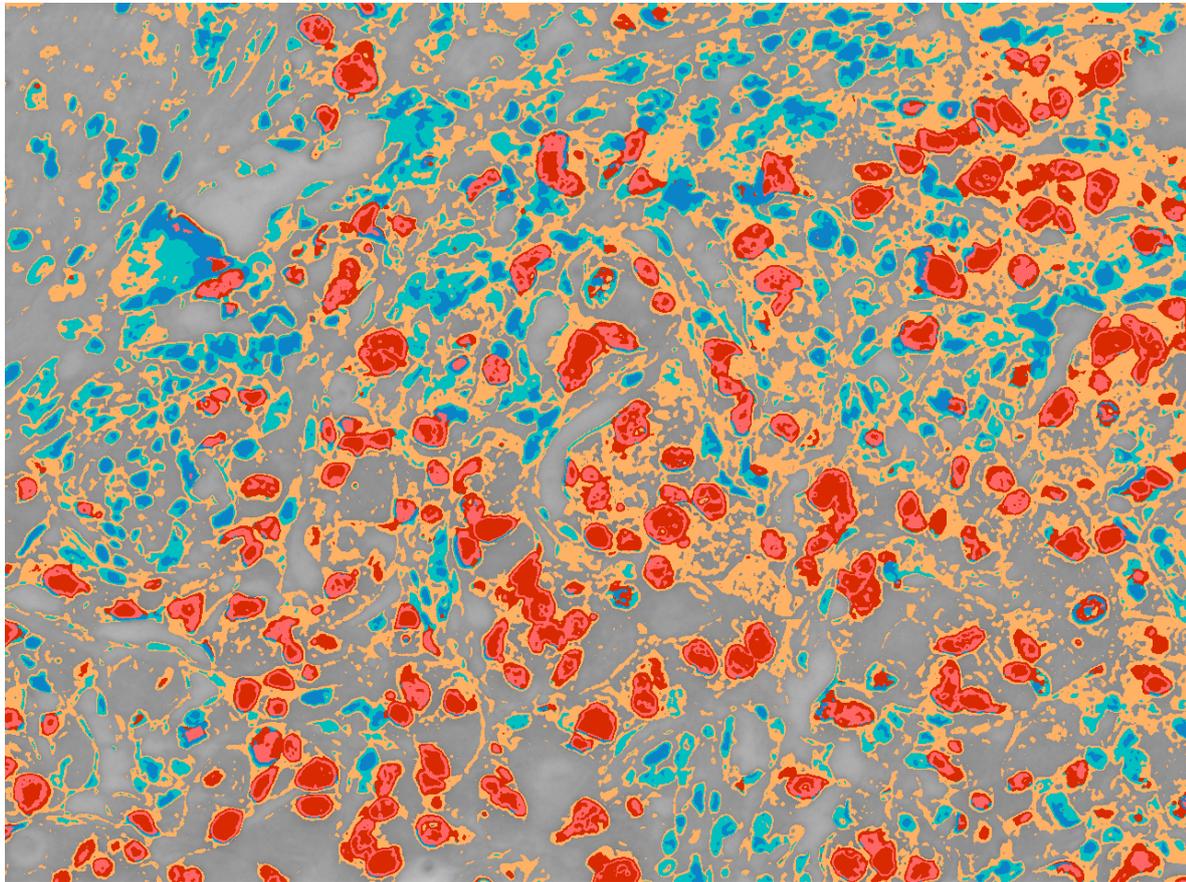


Figure A.7 Complete diagnostic pseudo-color map in Figure 5.2.3

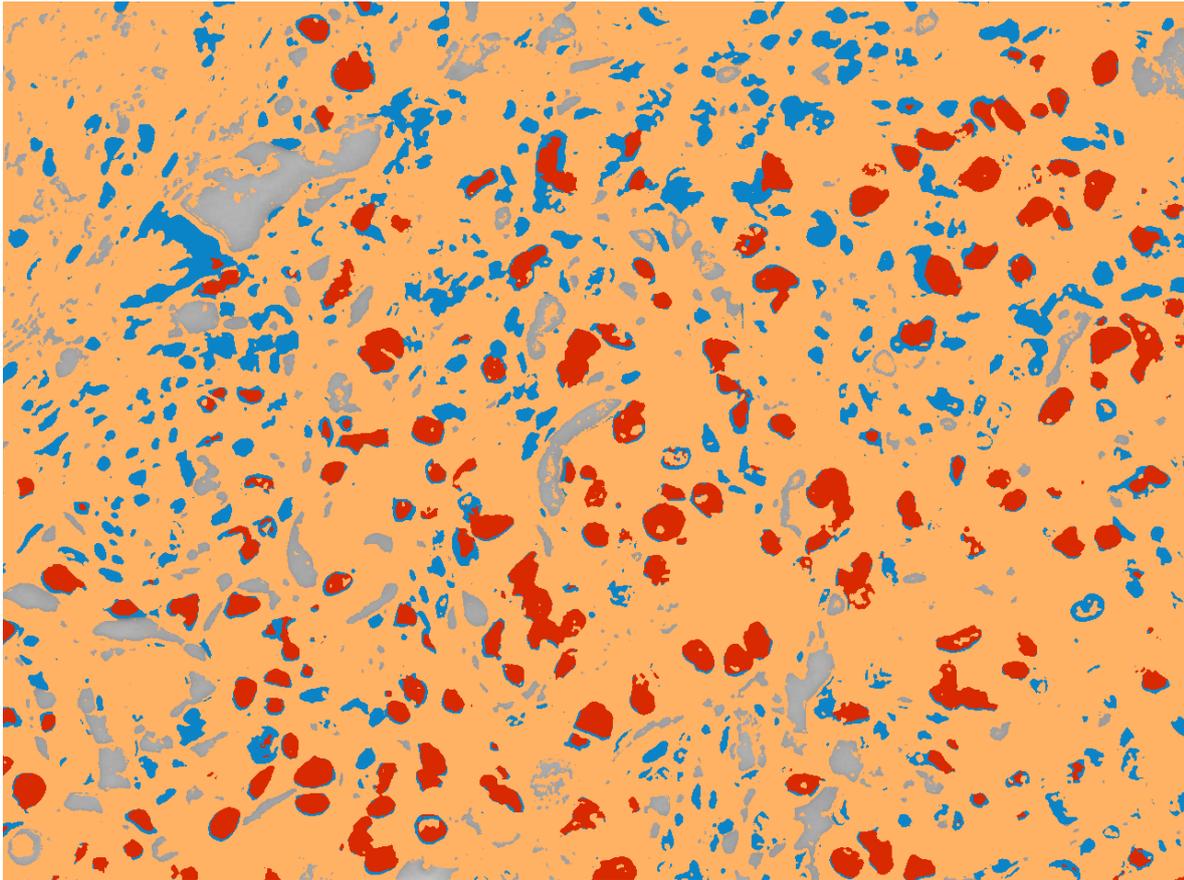


Figure A.8 Complete diagnostic pseudo-color map presented in Figure 5.2.4

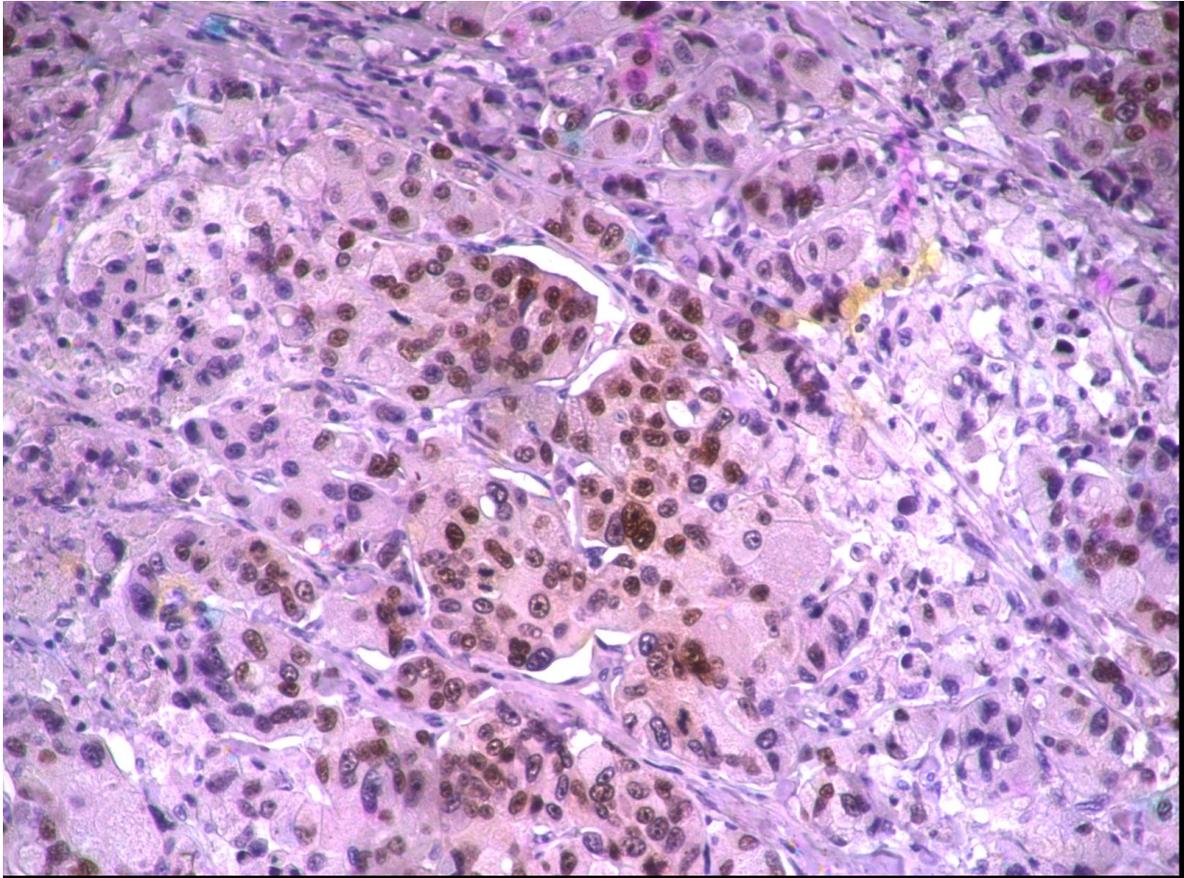


Figure A.9 Complete (normal zoom) shown in Figure 5.3.1

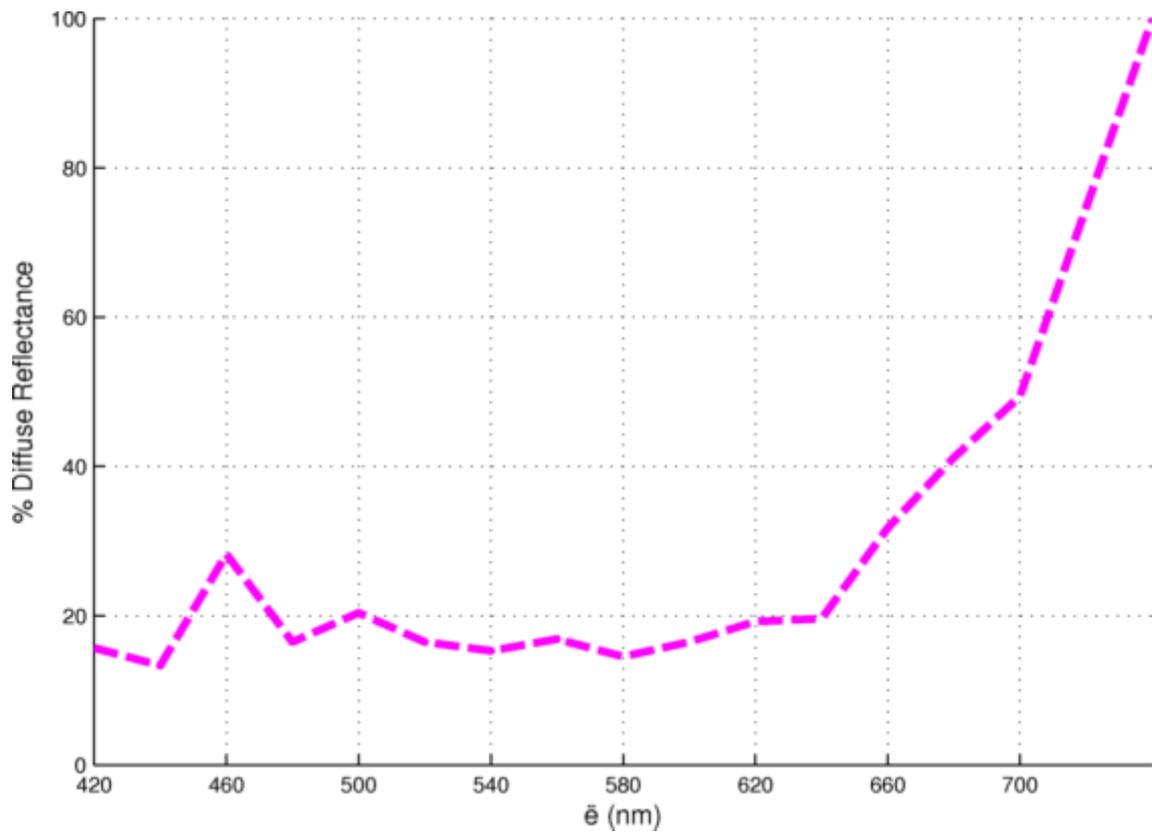


Figure A.10 Spectrum of cancer cells in Figure 5.3.1

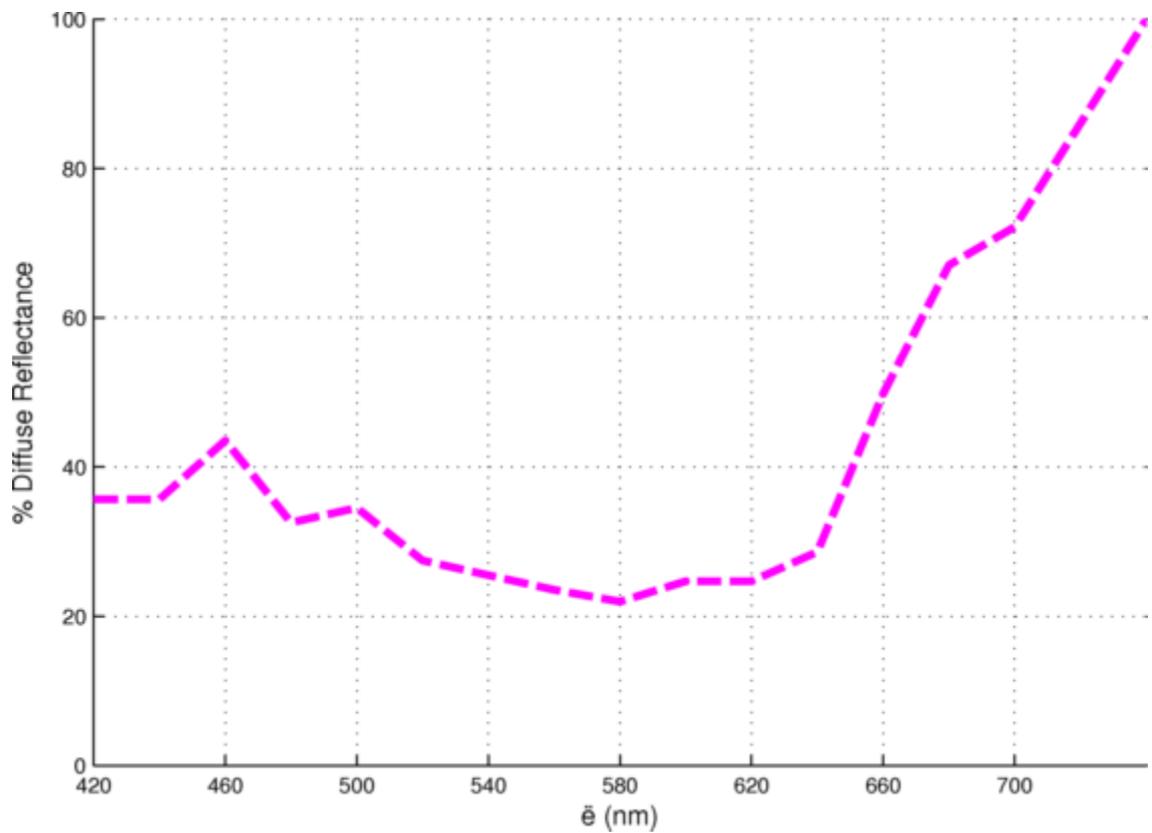


Figure A.11 Spectrum of healthy cells in Figure 5.3.1

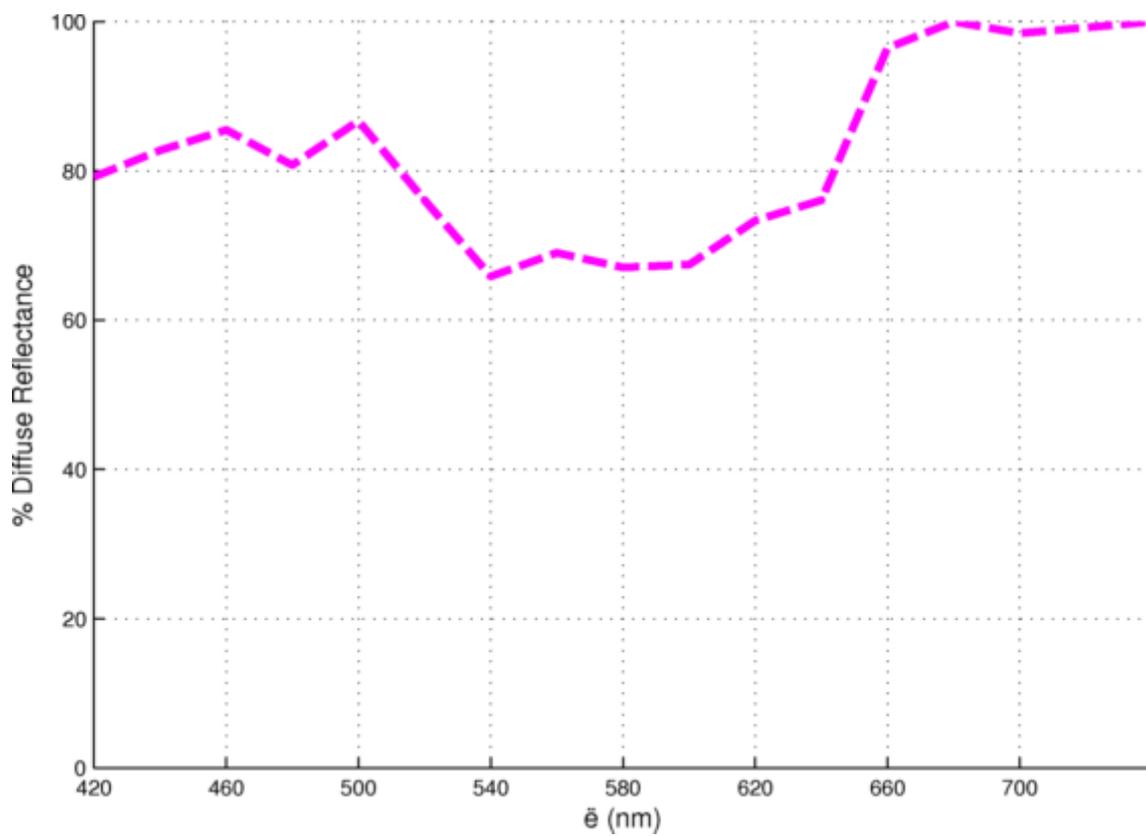


Figure A.12 Spectrum of non specific material in Figure 5.3.1

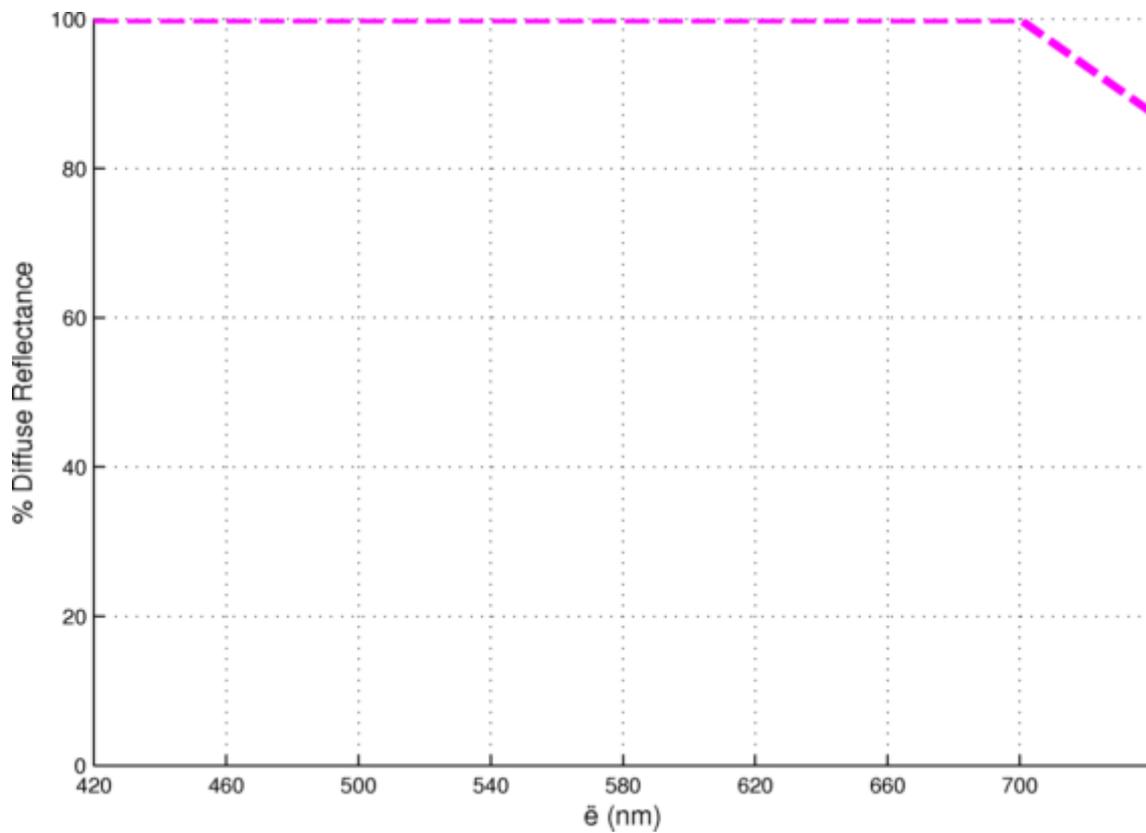


Figure A.13 Spectrum of background in Figure 5.3.1

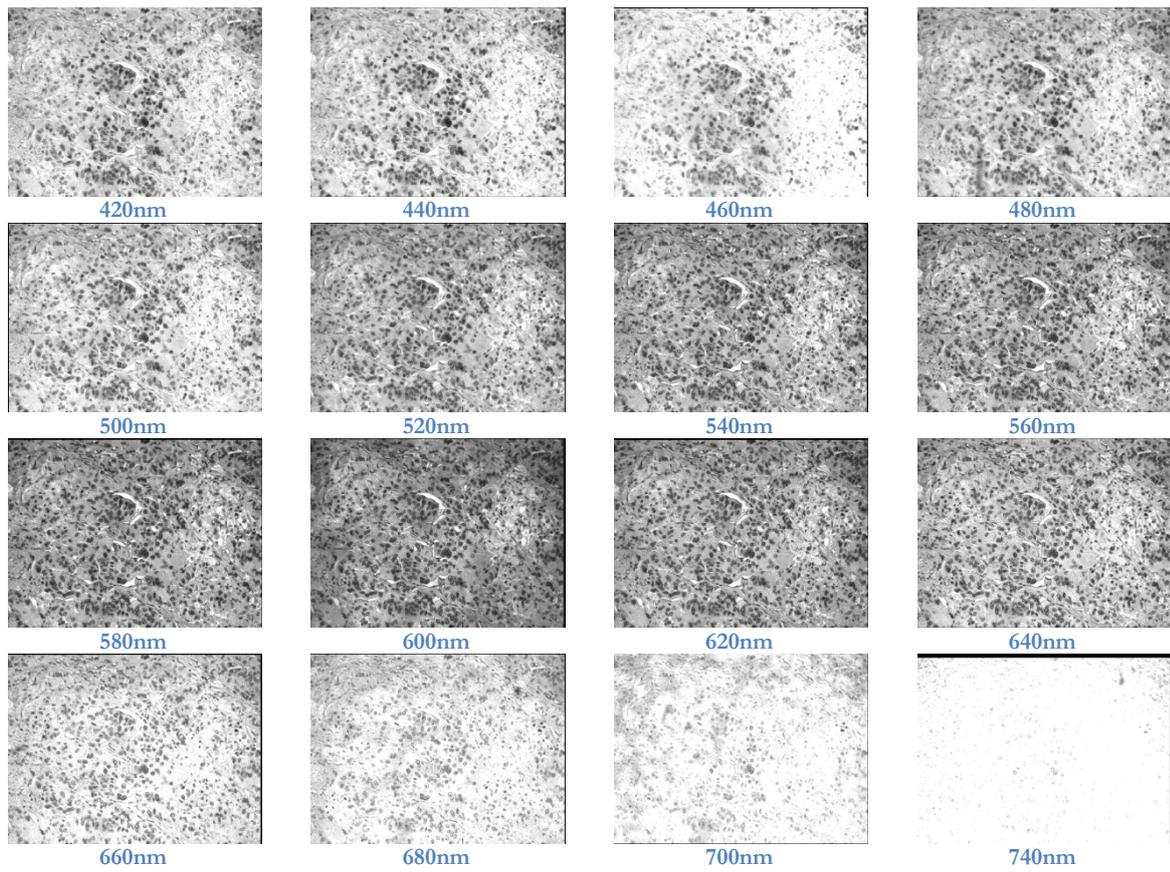


Figure A.14 Complete spectral cube presented in Figure 5.3.2

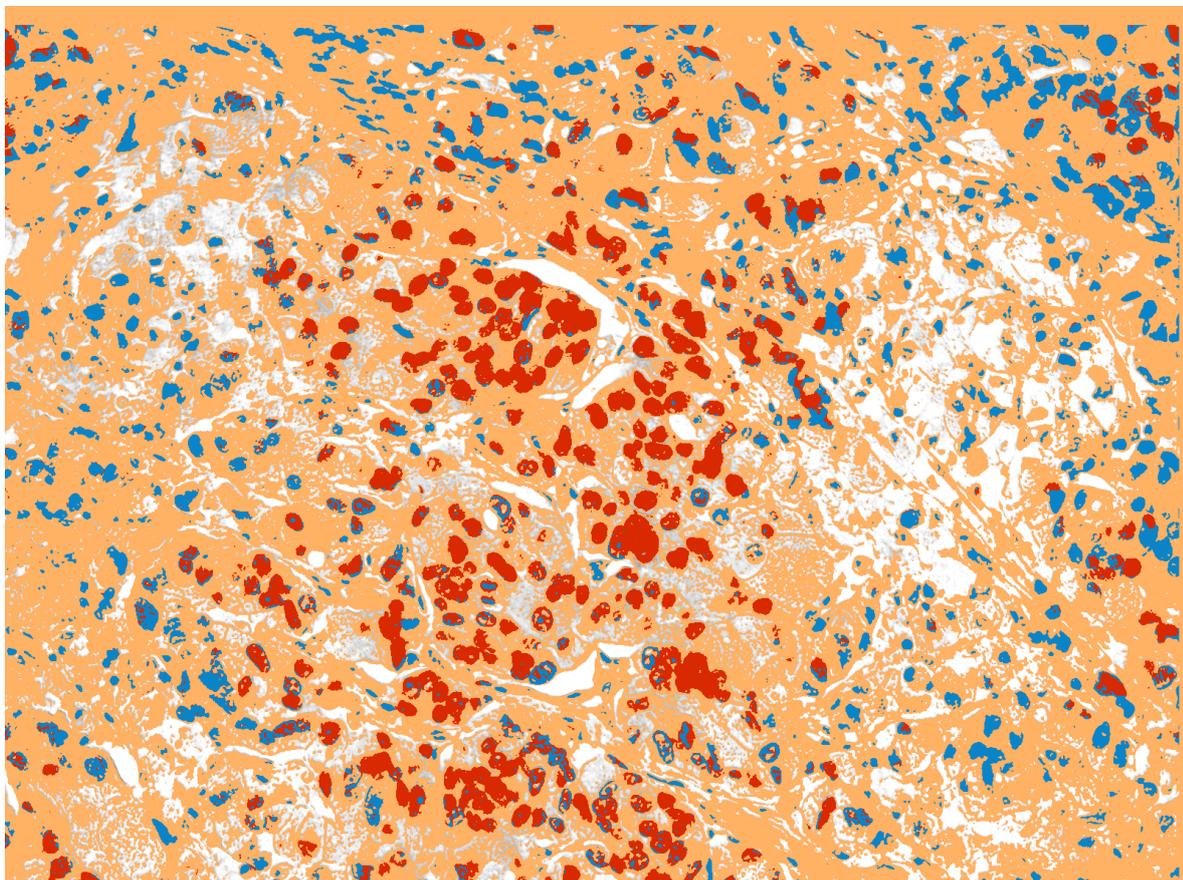


Figure A.15 Complete diagnostic pseudo-color map presented in Figure 5.3.3

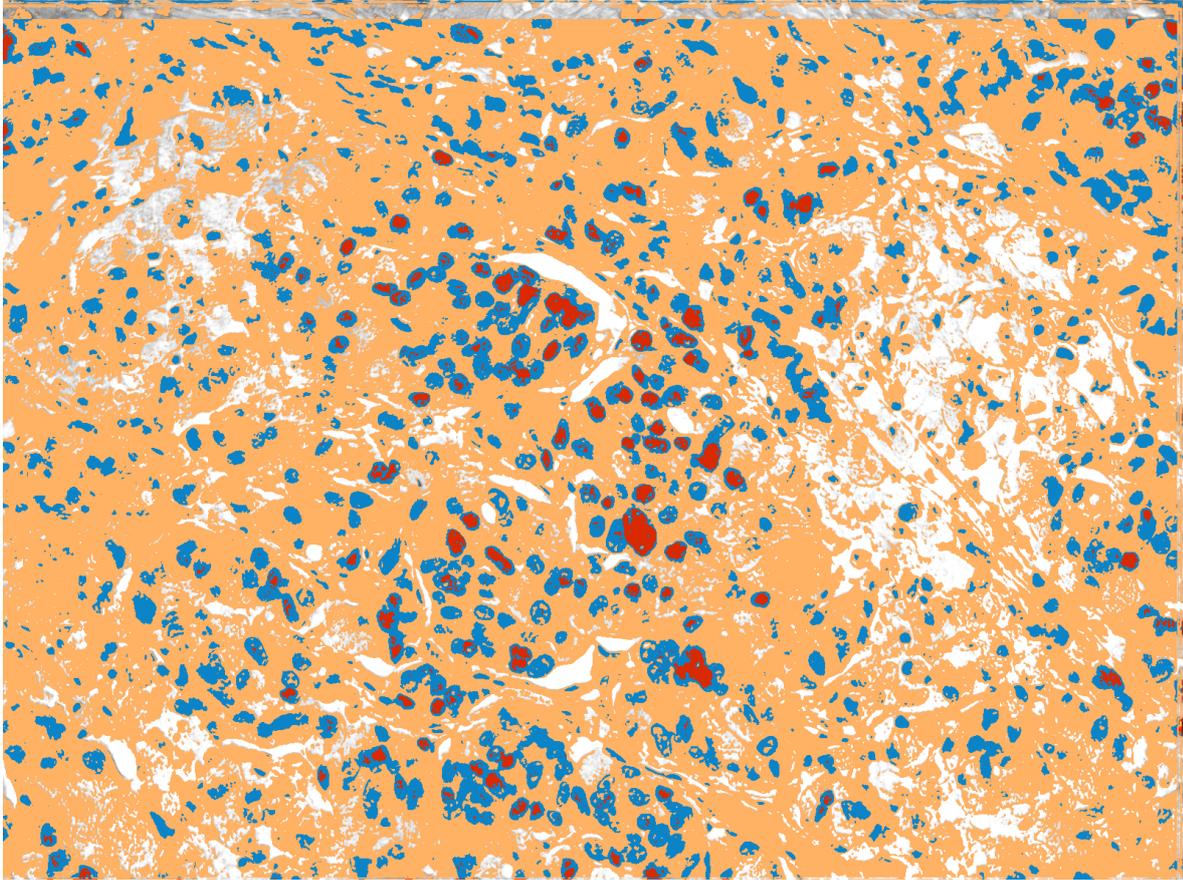


Figure A.16 Complete diagnostic pseudo-color maps presented in Figure 5.3.4

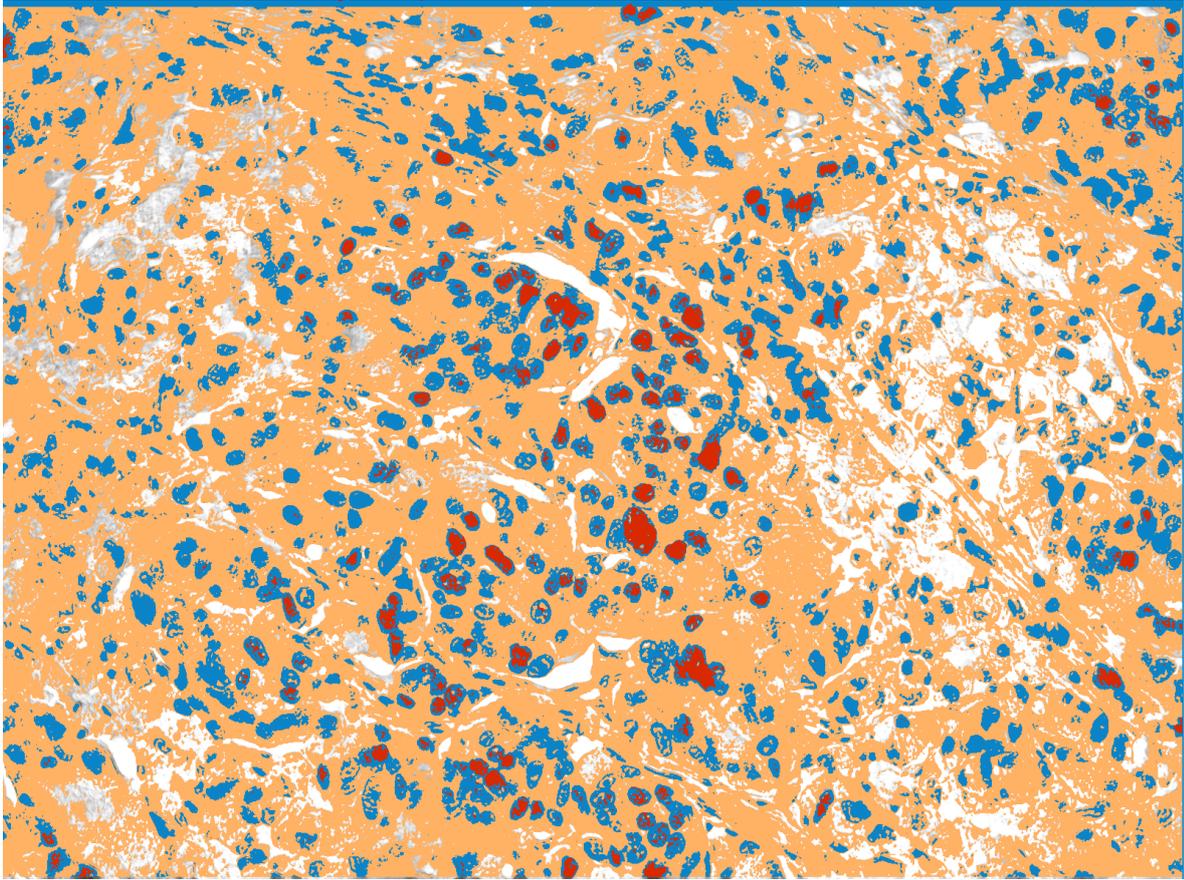


Figure A.16 Complete diagnostic pseudo-color maps presented in Figure 5.3.6