FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Spectral object reflectance estimation using smartphones

Miguel Cruz Fernandes

WORKING VERSION

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Luís Filipe Teixeira

February 8, 2018

Spectral object reflectance estimation using smartphones

Miguel Cruz Fernandes

Mestrado Integrado em Engenharia Informática e Computação

February 8, 2018

Abstract

Here goes the abstract written in English.

Resumo

O Resumo fornece ao leitor um sumário do conteúdo da dissertação. Deverá ser breve mas conter detalhe suficiente e, uma vez que é a porta de entrada para a dissertação, deverá dar ao leitor uma boa impressão inicial.

Este texto inicial da dissertação é escrito no fim e resume numa página, sem referências externas, o tema e o contexto do trabalho, a motivação e os objectivos, as metodologias e técnicas empregues, os principais resultados alcançados e as conclusões.

Este documento ilustra o formato a usar em dissertações na Faculdade de Engenharia da Universidade do Porto. São dados exemplos de margens, cabeçalhos, títulos, paginação, estilos de índices, etc. São ainda dados exemplos de formatação de citações, figuras e tabelas, equações, referências cruzadas, lista de referências e índices. É usado texto descartável, *Loren Ipsum*, para preencher a dissertação por forma a ilustrar os formatos.

Seguem-se umas notas breves mas muito importantes sobre a versão provisória e a versão final do documento. A versão provisória, depois de verificada pelo orientador e de corrigida em contexto pelo autor, deve ser publicada na página pessoal de cada estudante/dissertação, juntamente com os dois resumos, em português e em inglês; deve manter a marca da água, assim como a numeração de linhas conforme aqui se demonstra.

A versão definitiva, a produzir somente após a defesa, em versão impressa (dois exemplares com capas próprias FEUP) e em versão eletrónica (6 CDs com "rodela" própria FEUP), deve ser limpa da marca de água e da numeração de linhas e deve conter a identificação, na primeira página, dos elementos do júri respetivo. Deve ainda, se for o caso, ser corrigida de acordo com as instruções recebidas dos elementos júri.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vehicula lorem commodo dui. Fusce mollis feugiat elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu quam. Aenean consectetuer odio quis nisi. Fusce molestie metus sed neque. Praesent nulla. Donec quis urna. Pellentesque hendrerit vulputate nunc. Donec id eros et leo ullamcorper placerat. Curabitur aliquam tellus et diam.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum.

Acknowledgements

Aliquam id dui. Nulla facilisi. Nullam ligula nunc, viverra a, iaculis at, faucibus quis, sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur magna ligula, ornare luctus, aliquam non, aliquet at, tortor. Donec iaculis nulla sed eros. Sed felis. Nam lobortis libero. Pellentesque odio. Suspendisse potenti. Morbi imperdiet rhoncus magna. Morbi vestibulum interdum turpis. Pellentesque varius. Morbi nulla urna, euismod in, molestie ac, placerat in, orci.

Ut convallis. Suspendisse luctus pharetra sem. Sed sit amet mi in diam luctus suscipit. Nulla facilisi. Integer commodo, turpis et semper auctor, nisl ligula vestibulum erat, sed tempor lacus nibh at turpis. Quisque vestibulum pulvinar justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam sed tellus vel tortor hendrerit pulvinar. Phasellus eleifend, augue at mattis tincidunt, lorem lorem sodales arcu, id volutpat risus est id neque. Phasellus egestas ante. Nam porttitor justo sit amet urna. Suspendisse ligula nunc, mollis ac, elementum non, venenatis ut, mauris. Mauris augue risus, tempus scelerisque, rutrum quis, hendrerit at, nunc. Nulla posuere porta orci. Nulla dui.

Fusce gravida placerat sem. Aenean ipsum diam, pharetra vitae, ornare et, semper sit amet, nibh. Nam id tellus. Etiam ultrices. Praesent gravida. Aliquam nec sapien. Morbi sagittis vulputate dolor. Donec sapien lorem, laoreet egestas, pellentesque euismod, porta at, sapien. Integer vitae lacus id dui convallis blandit. Mauris non sem. Integer in velit eget lorem scelerisque vehicula. Etiam tincidunt turpis ac nunc. Pellentesque a justo. Mauris faucibus quam id eros. Cras pharetra. Fusce rutrum vulputate lorem. Cras pretium magna in nisl. Integer ornare dui non pede.

The Name of the Author

*“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”*

Isambard Kingdom Brunel

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

ADB	Android Debug Bridge
JDK	JAVA SE Development Kit
nm	Nanometers
IDE	Integrated Development Environment
XML	Extensible Markup Language
rms	Root Mean Square
JND	Just Noticeable Difference

Chapter 1

Introduction

Nowadays, measuring the spectral reflectance functions of surfaces is an essential requirement in many tasks. It is possible to analyse the measurement taken and get information that you could not get in any other way. The most intuitive task is probably object classification, e.g. making a statement about the age of food. Another prominent example is the creation of high-fidelity color images.

With the access to a spectrophotometer and a spectroradiometer provided by The Institute of Imaging & Computer Vision, in RWTH Aachen University, it is possible to take accurate measurements, later to be used in the development of thesis.

Having measurements done by proper equipment greatly reduces the mistake margin associated with the project, making possible a more accurate and trustworthy result.

1.1 Context

Object reflectances are usually measured in controlled environments. The reason is that when viewing an object, the spectral stimulus entering the measurement device will actually consist of two parts: the spectral object reflectance and the spectral power distribution of the illuminant. In order to separate both of them, the illuminant needs to be known.

The precise spectrum of the light reflected by a surface is usually measured using costly, high-end devices, such as spectrophotometers. Such devices are capable of directly measuring the wavelength dependent intensity of the incoming light. Loosely speaking, those devices allow a narrowband sampling of the incoming light. As a typical illuminant, perfect white is usually aimed at. In other words a broadband illumination.

A cheaper way to measure the object reflectance is to use narrowband illumination and a broadband sensing device. As an example, consider the surface of interest viewed by a monochrome camera, while the surface is illuminated with red light. The illuminant basically defines the sampling point of the object reflectance. Sweeping through the entire wavelength range equals to a complete sampling of the object reflectance.

Compared to the devices mentioned before, smartphones are able to provide an inexpensive, fast and practical solution. Metameric imaging is taken to consideration. A pair is called metameric if, under the same type of illuminant, they match in color. If the illuminant is changed then the pairs will have different spectral reflectances, meaning that they will not match in color anymore. The human eye, comparable to RGB cameras, uses three types of cone receptors to process spectral data over the visible spectrum, whose wavelength ranges from 380 to 780 nm, resulting in a three-channel color image. Being able to obtain this data allows for calculations to be performed in order to reconstruct spectral reflectances.

1.2 Motivation and Goals

While the measurement process itself is well known and straightforward, the question arises how well this task of object reflectance measurement can be performed with standard consumer hardware as used in smartphones. This will be where this dissertation is addressed at.

First, an application is created in order to manipulate the smartphone to illuminate, using the device's screen. To better understand which output we are getting, it is crucial that the colors presented are measured first, in order to preform a calibration, since machinery and the human eye perceive these colors in different ways.

Second, having the measured colors allows for a reconstruction of an object reflectance. There are different ways of doing this, whose results will be compared. Applying these reconstruction algorithms to a data set of objects reflectances allows for a more accurate comparison of the results.

To be finished...

1.3 Structure of the thesis

Besides the introduction itself, this thesis contains 3 more chapters. On chapter 2 we have a description of the bibliographic review and related projects. Chapter 3 refers to the implementation. This includes all the android development, as well as the device's screen illumination calibration. Chapter 4 refers to the results, future work and the work plan.

Chapter 2

Bibliographic Review

In this chapter the Bibliographic Review will approach concepts, techniques and technology currently used when it comes to spectral object reflectance. We will start by defining spectral reflectance and how to measure it accurately, followed by the defenition of object reflectance and the algorithms used to calculate it.

2.1 Spectral Reflectance

Different surfaces reflect or absorb a radiation in different ways. The reflectance properties of a given object or surface depend on the material it is made of, the surface roughness and the circumstances it is measured in. By analysing the spectral signatures, it is possible to identify different surface features or materials. These can be visualised in spectral reflectance curves as a function of wavelengths.

2.1.1 Usability

The spectral signatures have many different uses, namely in agriculture and medicine, for instance. Figure 2.1 shows typical reflectance curves of water, vegetation and soil. Healthy vegetation has a smaller reflectance in the visible spectrum, which then increases when it starts nearing the infrared. As a practical example, stressed vegetation or vegetation in different life lengths can also be identified due to its significantly lower reflectance in the infrared region (Figure 2.2). This data can be used to differentiate said vegetation in order to improve farming activities (for commercial or personal use or simply for paisagistic development).

2.1.2 Electromagnetic Spectrum

Light is radiation in the form of electromagnetic waves that make vision possible to the human eye [OR06]. Electromagnetic radiation can be classified by its wavelength or frequency, as shown

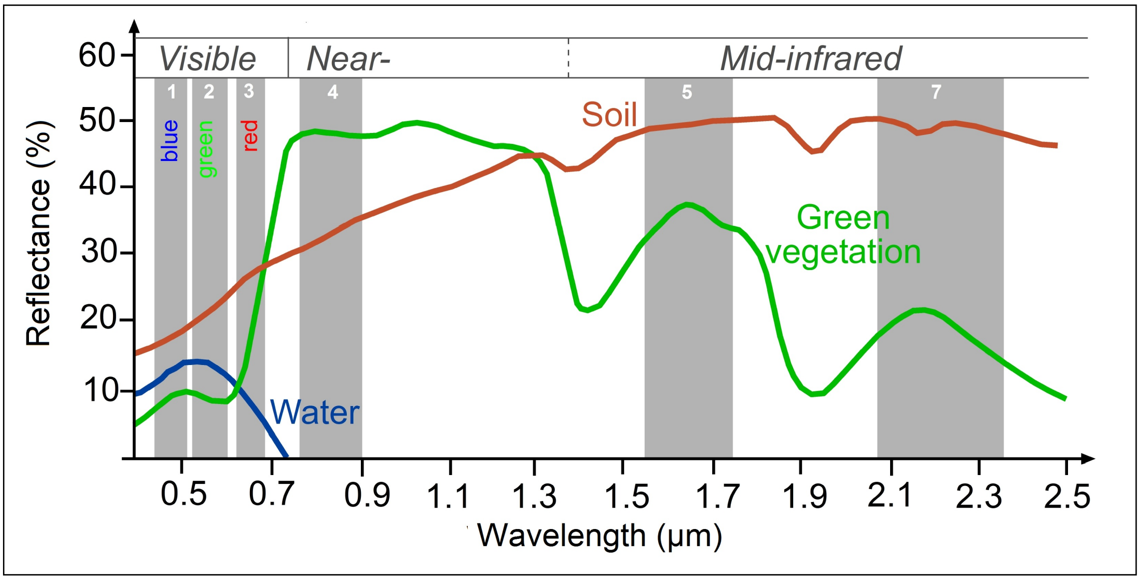

Figure 2.1: Spectral signatures of water, vegetation and soil

in Figure 2.3.

The spectrum covers electromagnetic waves with frequencies from below one hertz to above 10^{25} hertz. The frequency range is divided into separate bands (starting in low frequency): radio waves, microwaves, infrared, visible light, ultraviolet, X-rays and gamma rays. Exposure from high ultraviolet to smaller wavelengths is considered to be a health hazard, because of the chemical reactions that can occur, causing DNA damage and cancer, for instance. The radiation from visible light or lower frequencies cannot cause such effects.

2.1.3 Spectral Object Reflectance Reconstruction

The objective of the reconstruction is to make estimations from a low-dimensional data into high-dimensional data. Considering that a signal might have been corrupted by noise, various algorithms will be used to try to filter out the noise from the given signal. This way, we end up with an estimation that might just be comparable enough with the original reflectance, allowing us to reach a conclusion on how efficient the hardware from the smartphone can be, for this use.

2.2 Measurement tools

In this section the tools provided by the institute will be described. Since the final objective of this thesis is the comparison of our smartphone device to higher cost devices, it was crucial to have access to such.

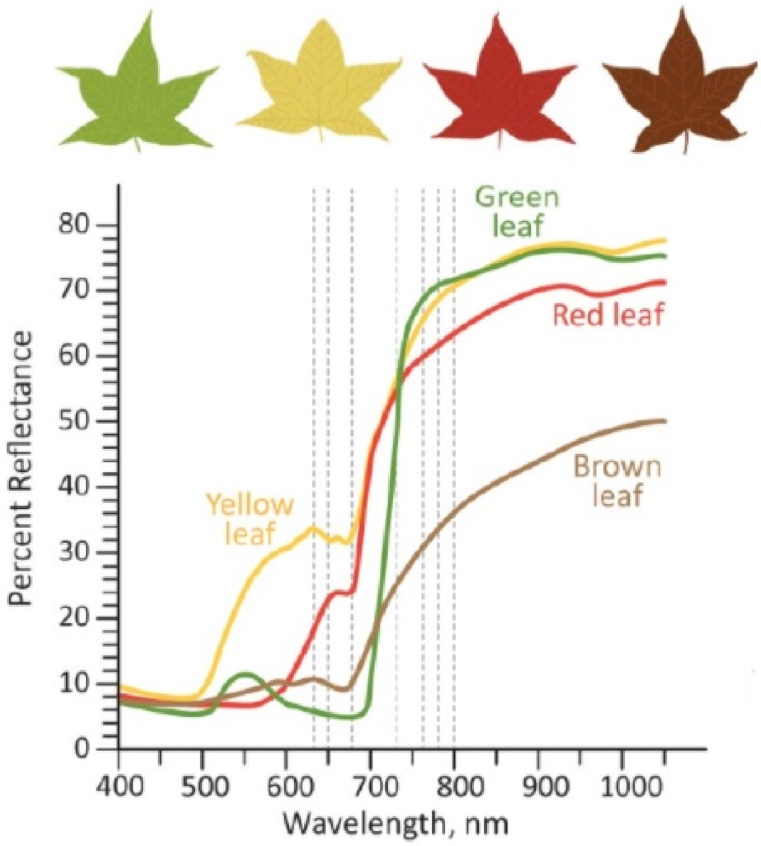

Figure 2.2: Spectral signatures of different vegetation

2.2.1 Spectroradiometer

A spectroradiometer is a device designed to measure the spectral power distribution of a light source. In the current case, that light source is an illuminated smartphone screen. The output provided by this device, controlled with Matlab, is utilized to create a colour graph in a given wavelength.

Any spectroradiometer will integrate four basic units [Ltd14]:

1. The input optics, to gather the electromagnetic radiation from the source.
2. A monochromator, to separate the radiation into its component wavelengths.
3. A detector, to measure the radiation at each wavelength.
4. A control and loggin system, to define gathered data and store it.

Regardless of how technically good the system is, the produced data is only as good as the calibration of the instrument and the measurement conditions. This means that a lot of factors contribute for a better measurement, but not all can be controlled, such as noise from the detector, internal electronics, humidity and temperature variations, for instance.

These devices are used in different applications, such as plant research and development and LED measurement [Sci15].

2.2.2 Spectrophotometer

A spectrophotometer is a device designed to measure the amount of light reflected or absorbed from a surface or object, resulting in a function of wavelength.

The measuring process is as follows:

1. A light source in a certain wavelength shines onto the surface.
2. The surface reflects or absorbs light.
3. The device detects how much of the incident light is reflected from the surface.
4. The device converts how much light the surface reflected into a numeric value.

This allows for a graph to be drawn, in the wavelengths that the light source provided. As an example, if we take the visible light spectrum and measure every nanometer (nm) that is reflected/absorbed, it is possible to determine the intensity of each wavelength in the resulting graph.

With this in mind, there are several applications for this device, such as medicine or agriculture [C.15], among others. Since different compounds absorb best at different wavelengths, it is possible to identify them, as in Figure 2.4, for instance. Nitrophenols are poisonous [Boo00] and contaminate soil near former explosive or fabric factories and military plants. Identifying these substances allows for a solution to be developed based on the data, for healthy and safe soil usage.

2.3 Software tools

2.3.1 Android vs iOS

Although the iOS platform has a solid environment when it comes to development, Android was chosen for this project. Having close to no background in smartphone application development was also part of this choice. Android supports the developer with very useful tools such as the Integrated Development Environment (IDE) named Android Studio. Knowing that this has been a replacement for Eclipse and that it utilizes Java, which I was more familiarized with, finalized the decision.

2.3.2 Android Studio

Android Studio is the official IDE for Google's Android operating system. Android Studio supports Java and the layouts are designed in Extensible Markup Language (XML), which are very easy to manipulate due to its' drag-and-drop visual editor. It is a tool with great flexibility and support, providing simple tutorials and general information to new developers [Goo13].

2.3.3 MATLAB

MATLAB was developed by MathWorks which combines a desktop environment with a programming language that expresses matrix and array mathematics [Mat84].

Since most of the provided algorithms were written for MATLAB, as well as the control for the measurement tools, it is essential to utilize this software, not only for its' flexibility but also for the easier communication with Java.

MathWorks helps users allowing files to be shared for common usage, providing support and accurate data such as spectral and *XYZ* color functions [Mat10] and color properties toolboxes [Wag07], which prevent errors in large but precise calculations.

Three different estimation algorithms were used for this work:

2.3.3.1 Wiener Estimation

2.3.3.2 Linear Estimation

2.3.3.3 Principal Eigenvectors

2.3.4 Java

Java is a concurrent, class-based and object-oriented programming language [Gos15]. It is very useful for multiple platform use, since it doesn't require recompilation, as long as said platform supports the language.

It gives the user access to different libraries, two of which had a bigger role in this project.

2.3.4.1 MatlabControl

MathWorks supports calling and using Java objects from Matlab, but not calling Matlab commands from Java. MatlabControl allows the user to call Matlab commands from Java.

It is essential for the measurements control as well as managing results, considering that it is simpler and wiser to use minimum number of different programming languages. It could be possible to communicate with Matlab through Python, for instance, but it would disregard the previous statement, since Python would not be used for anything else.

2.3.4.2 JFreeChart

JFreeChart is a Java chart library that allows the display of charts or graphs [Gil00]. When in need of visualizing multiple results, it is easily accessible to work with. It has an extensive feature set, of which the line graph proves to be the most useful for the visualization and comparison of results.

2.4 Conclusions

Overall, the thesis mostly utilizes Java and Matlab. The interaction between the two acts as a user interface to facilitate obtention of faster results. It gives the possibility to either analyse a small amount of the data, as well as much larger amounts, with ease.

Minimum human interaction with Matlab is required, but it is also possible to deeply analyse results, if desired, based on the fact that the Matlab session is opened and available for use during the entire measurement/calculations process.

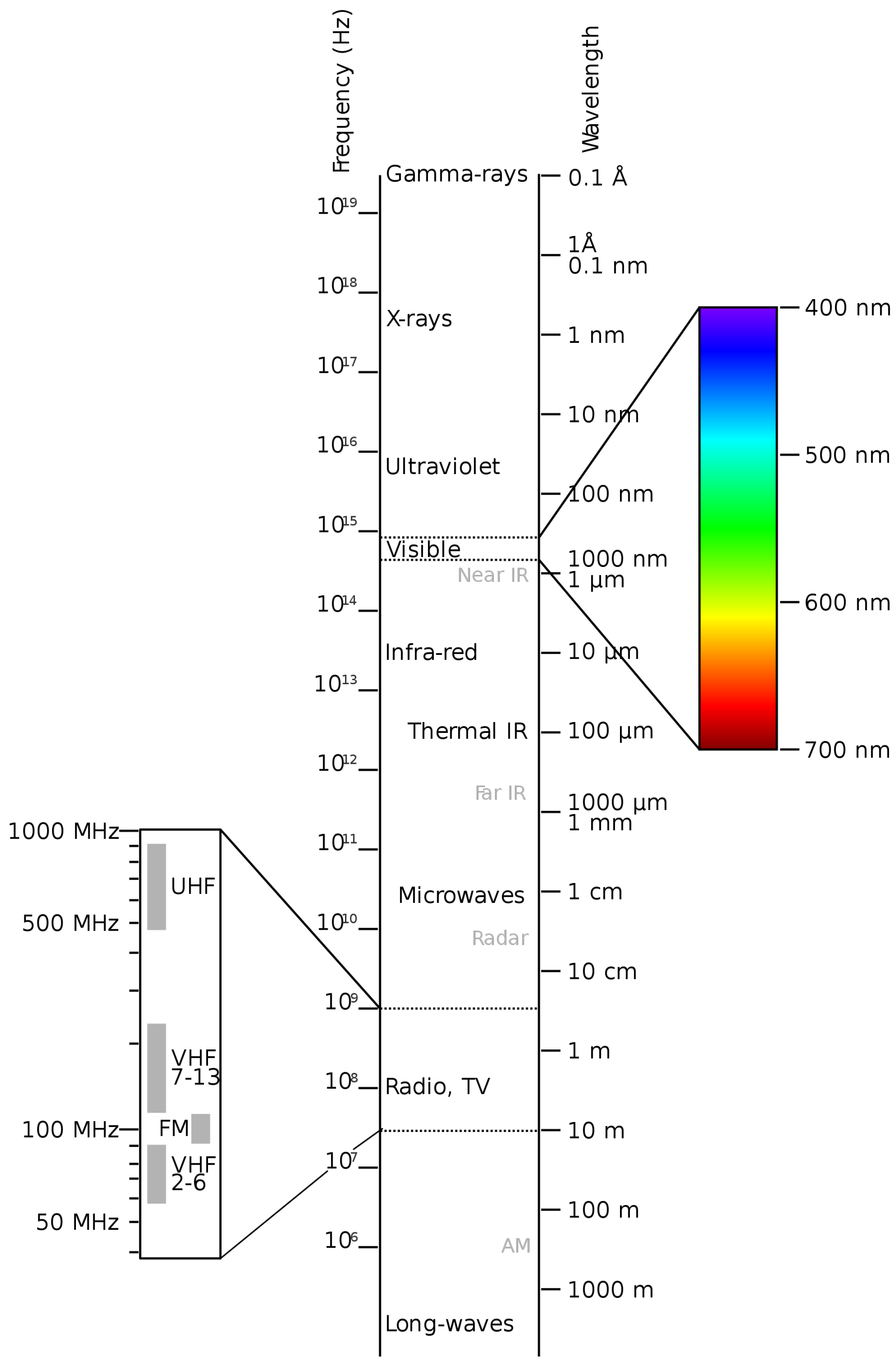

Figure 2.3: Wavelengths of electromagnetic radiation and light

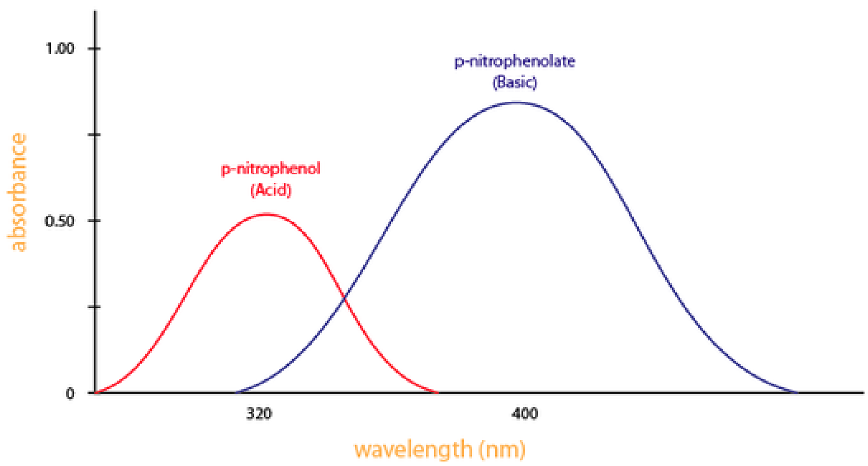

Figure 2.4: Absorbance of p-nitrophenol and p-nitrophenolate

Chapter 3

Implementation

This chapter is divided into three different parts: the Android application development, the object reflectance estimation and the evaluation.

3.1 Android application development

The main objective of the application is to be able to act as the illuminant and take a picture of the object reflecting the incident light. The smartphone is held with its display towards the desired surface, having the screen to be fully illuminated with predefined sets of chromatic colours. After the display has been set to a specific colour, a picture can be taken, which is then automatically saved to a specific folder in the smartphone's memory.

3.1.1 Initial mockups

Initially, some simple mockups were created to try to understand what was important for the application to contain.

The initial idea was to have a main screen, as presented in Figure 3.1, where you can select predefined colour sets to go through, as an illuminant. There are options for 1, 3, 6 or 9 distinct colours, which a user can select and define. There is also a gallery and a transfer button. The first one, to browse through the taken pictures and the second one to be able to select the desired pictures to be able to transfer it to a computer.

Figure 3.2 shows the outcome of pressing the "9" button in the main screen. There, the user can visualize the colour hexcodes defined for the picture taking process, as well as start said process, by clicking the "Begin" button.

Although the current prototype (which will be thoroughly described next) is not working as such, pressing the colour hexcodes buttons would lead the user to another screen that would allow to define the exact colours to be used, as well as their showing sequence, as seen in Figure 3.3.

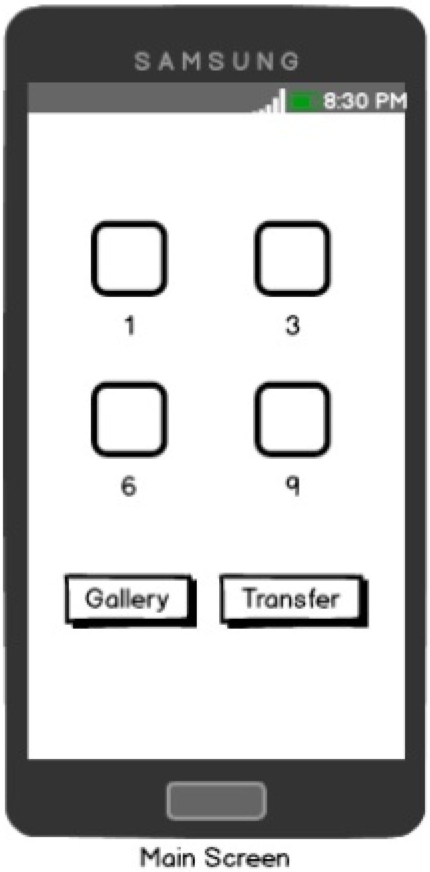

Figure 3.1: Main Screen Mockup

In this screen, not only can the colour be selected in the RGB colour model, but also as a hexcode, allowing a preview of said colour. Upon confirmation, the user is returned to the previous screen, enabling the definition of the remaining colours.

The gallery (Figure 3.4) and transfer screens (Figure 3.5) are almost identical, with the only difference being the selection on the transfer screen, since it enables the user to select the desired pictures, before transfering them.

The main part of the application, which is the screen serving as illuminant, is only showed in the prototype section, since it only contains the full screen illuminated with the selected colour.

A last mockup that was designed is for the screen immediately after the picture taking process is finished.

Given that, for instance, the "9" button is selected in the main screen (Figure 3.1), followed by the beginning of the activity, only nine colours would be available. This means that when changing to the next colour, after going through the first nine, which does not exist, the photo process is considered finished. We are finally presented with a results screen, as shown in Figure 3.6. In said screen, the user is able to immediately transfer the pictures taken, repeat the whole process or simply confirm the end of process, which will result in the main screen being presented again.

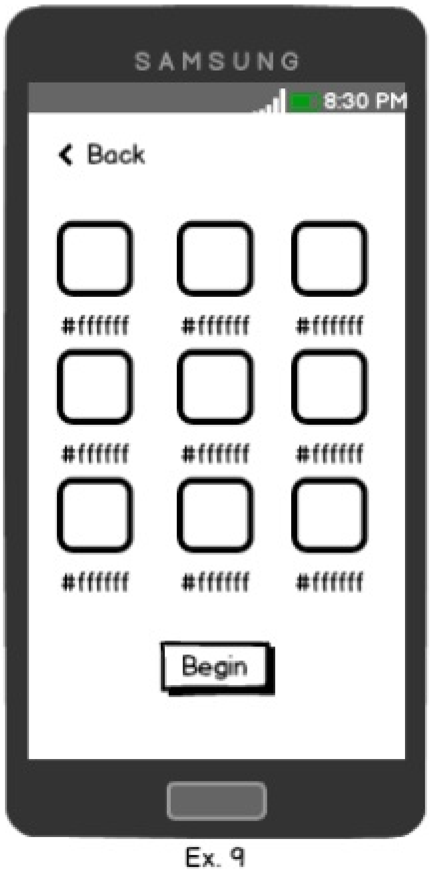

Figure 3.2: Example 9 Mockup Screen

3.1.2 Functional prototype

The prototype started with the mockups as inspiration, but with some adjustments.

From the main screen, as seen in Figure 3.7, the major differences to the original idea are the "Calibration" button and the "Presets" button. The first one is described in the next section, but the "Presets" button is an extension the application.

Instead of choosing every colour, one by one, like in Figure 3.2, this button leads us to the preset activity screen (Figure 3.8). Here the user can either select, edit (Figure 3.9) or delete a previously created preset, as well as create a new one, by clicking the "Create preset" button (Figure 3.10). The main advantage of this system, compared to the original mockup, is that one can paste the colour's hexcodes, instead of defining them one at the time. For measurements containing hundreds of different colours, this method proves to be much more efficient. For file reading purposes, it is requested that the user writes down the colors in the format described on the top of the screen.

When the user wants to use a preset, a new activity begins, the brightness is brought up to 100% and the front camera becomes active and visible in the screen.

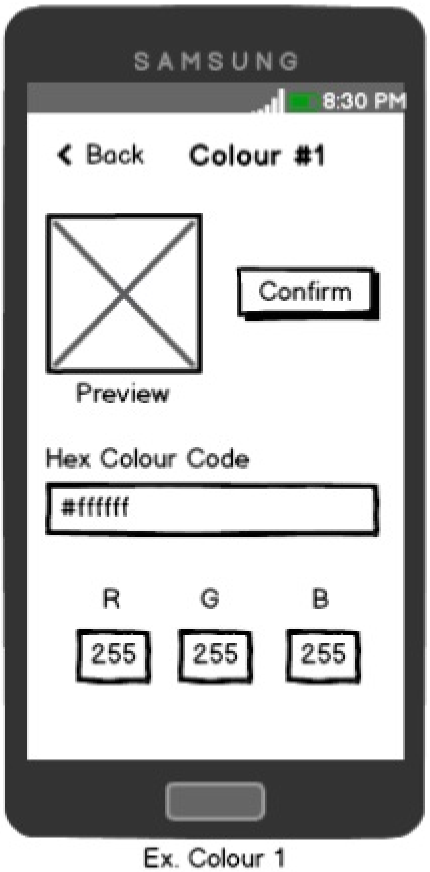

Figure 3.3: Example Colour 1 Mockup Screen

To avoid moving the smartphone as much as possible, the original idea revolved around an automatic timer, that would change to the next colours and take a picture in between. This idea was discarded for two main reasons:

- Efficiency, since the user may or may not want to go through the whole preset list. In case there are only a few colours that the user might want to repeat, it is possible to skip unwanted ones, saving time and phone memory. Obviously, it would be fairly easy to create a new preset, but slower in the long run.
- Freedom of usage, since the user may want to adjust settings in between pictures, such as room lighting and smartphone positioning, for instance.

Instead, the manipulation of the picture taking and colour changing process is done using the up and down volume buttons. With the click of the "Up" button, the screen changes to the first colour defined, identifying it by briefly showing the hexcode associated with it, as demonstrated in Figure 3.11. The user is then able to take a picture by clicking the "Down" button. A big advantage of this method is that even tho the picture cannot yet be previewed, there is the opportunity of

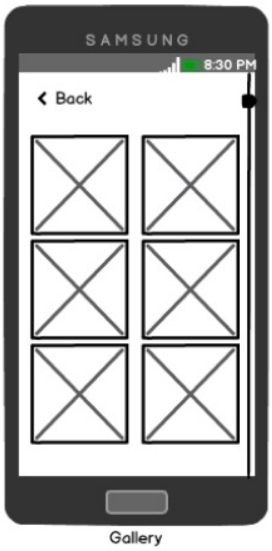

Figure 3.4: Gallery Mockup Screen

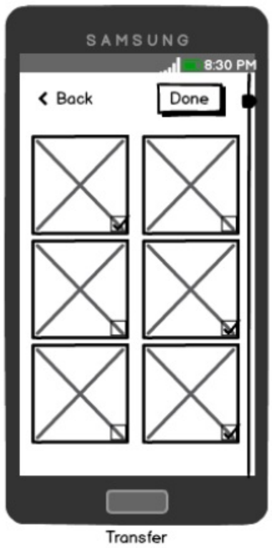

Figure 3.5: Transfer Mockup Screen

taking as many pictures as one might want. When satisfied, simply click the "Up" button again to change to the next picture and so on and so forth, until there are no more colours to change to, which leads us back to the main screen.

3.2 Device's Screen Calibration

In order to be able to get more reliable results, it is necessary to have some sort of screen calibration, to be able to analyze the exact colours that are being used.

The reason why this is needed is because we are utilizing cheap hardware, which is not trustworthy. As observable in the results of this calibration, the measurements have errors. As we will demonstrate, just because we define a specific colour to be used in the screen, which may appear perfect to the human eye, it doesn't necessarily mean that it is.

This section is divided into three subsections, where the first one focuses on the connection and activity among the smarthphone, computer and spectroradiometer. The second subsection will focus on the measurements taken and their visualization with the jfreechart library. The third subsection will be a brief analysis and conclusion based on the graphical results.

3.2.1 Connection

First, it is important to know how the connection was implemented. We have three different devices, that somehow need to communicate with each other. Each device has different programs running in them. Let us describe them one by one.

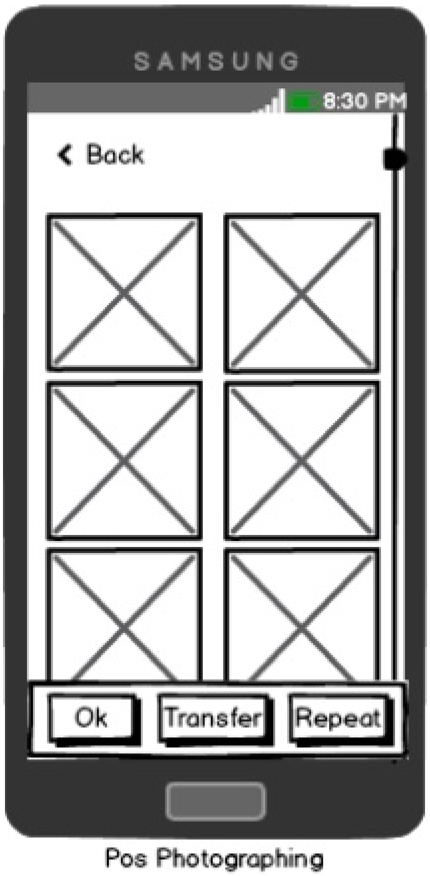

Figure 3.6: Results Mockup Screen

3.2.1.1 Computer (Host)

There needs to be a host throughout the whole communication. A JAVA program was developed to manage this communication. The software communicates directly with the Spectroradiometer and with the smartphone, since these two never communicate with each other. Both of the devices are connected via USB port to the host.

There is some other software that enables the connection which is needed for this process, such as Android Debug Bridge (ADB) and JAVA SE Development Kit (JDK), described in the Bibliographic Review (chapter 2). There should also be a file previously created with the colours that need to be measured.

The host[1] creates a socket to be able to communicate with the smartphone device, by sending and receiving messages. There is a "Connect" button in the smartphone calibration screen, which after pressed listens for clients. After connecting through the ADB, the host opens a new Matlab session, to begin the measurements. The full measurement process works as such:

1. The smartphone initializes the connection and waits for the JAVA code to begin.

[1] We use the term "host" since, in the end, the computer is in charge of the commands. In reality, the smartphone is the host, since it is there that the connection is initialized.

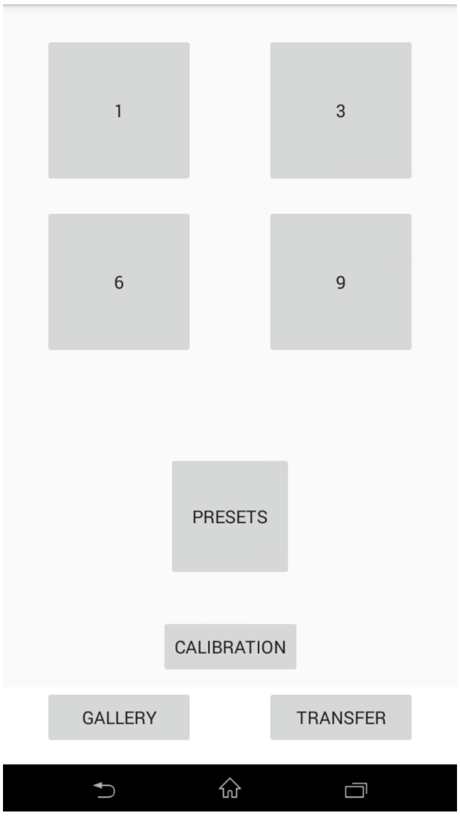

Figure 3.7: Prototype Main Screen

2. The host completes the connection, openning a new Matlab session and loading the code to manipulate the Spectroradiometer.

3. While the host has lines with colour hexcodes, it sends that same line to the smartphone.

4. The smartphone changes the screen background to the colours recieved and informs the host about it.

5. Utilizing MatlabControl library, the host sends the commands necessary to take a measurement.

6. The Matlab code replies with a "Measurement taken" type of message and we go back to number 2, until the end of the colours file has no more colours to read from.

7. When there are no more colours to be read from the colours file, both the Matlab session and the socket with the smartphone device are closed.

The full JAVA file (ScreenCalibration.java) for the communication can be found in the appendix.

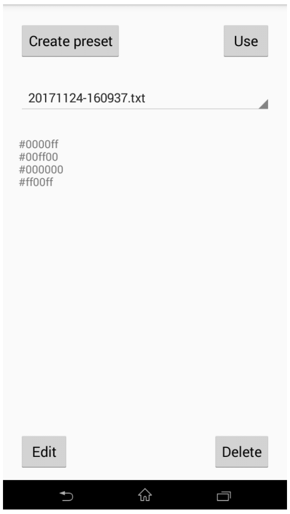

Figure 3.8: Prototype Preset Activity Screen

Figure 3.9: Edit button on Preset Activity Screen

3.2.1.2 Spectroradiometer

The spectroradiometer available already had Matlab code provided for its utilization. It allows measurements to be taken, on command, providing a long list of data that can later be transformed into graphs, for better understanding and visualization.

For the calibration itself, it is important that the device only measures when told so, meaning when the smartphone is displaying the desired colour. This interaction is described in the last sub-subsection.

The MatlabControl library allows a JAVA file to send commands to a Matlab program. For this specific program, the command lines used are:

- cd('Y:\\Spectroradiometer') - which opens the directory where the program is;
- addPathWithSubpathes - which readys the correct program for usage
- comPort = 'COM3'; - which has to be set manually, according to the port that the device is connected to
- CS2000_initConnection(comPort); - which initializes the connection
- CS2000_terminateConnection(); - which terminates the connection

Specific to measurements, since the host needs to wait for the measurement to be taken, the function returningEval is used. Each time a measurement is requested it is done as in Listing 3.1, where "proxy" is the name of the Matlab session and "res" contains a 401x1 double, which is then saved into a new text file, named after that specific measurement.

3.2.1.3 Smartphone

The smartphone device part just needed some minor adjustments to work its role in the calibration process.

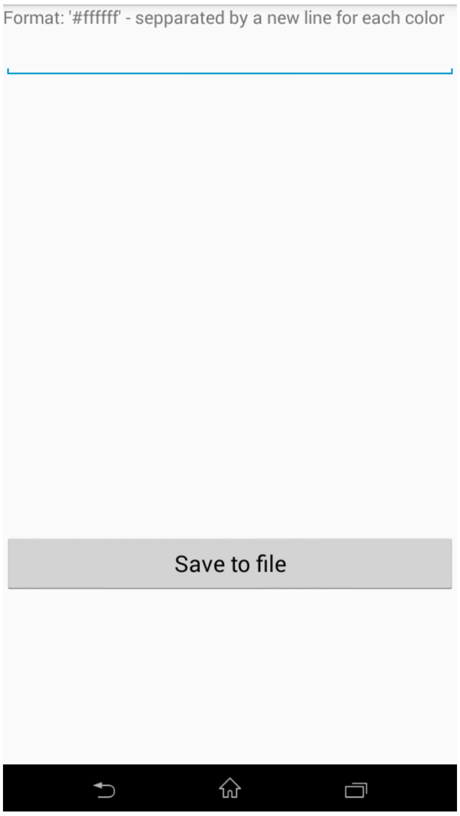

Figure 3.10: Prototype Create Preset Activity Screen

It initializes the connection on button click and listens for 60 seconds to see if a client connects. If no one connects, it launches a timeout. If the connection is successful, it sends a "READY" message to the client and waits for the colours to be send, so it can change the background. After it receives an "END" message, which indicates there are no more colours to change to, it replies with "ENDED", returning to the main screen.

3.2.2 Measurements Visualization

After executing ScreenCalibration.java and measuring the colours from our selected text file, a folder with the same name is created in the same directory, which contains a text file for each colour measured.

```
proxy.eval("[message1, message2, cs2000Measurement, colorimetricNames] =
    CS2000_measure();");
res = proxy.returningEval("cs2000Measurement.spectralData", 1);
```

Listing 3.1: Commands to take and save 1 measurement

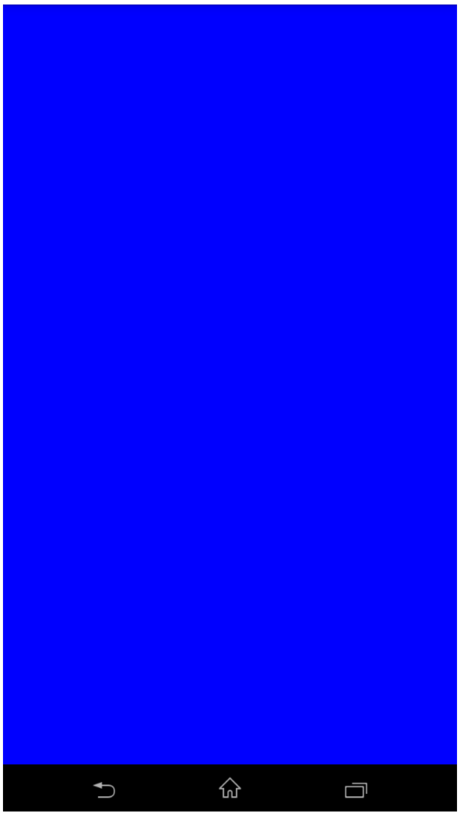

Figure 3.11: Example of a preset being used

Considering that we have measured, for instance, *#ff0000*, *#00ff00* and *#0000ff*, three text files with the same name as the colour were created, with 401 values in each, which covers the visible spectrum, from 380 to 780 nm. Having such files allows us to create a graph for better visualization of what the spectrophotometer has measured, to be able to compare to the ideal wave that that specific colour should have. The results will be analysed in the next subsection.

GraphMaker.java works as an extension of the ScreenCalibration.java, since it reads from the folder containing the measurements. With the help of JFreeChart library, we open each measurement text file and create a JPEG graph of each colour (following the example mentioned above, three graphs would be created). The library offers many different choices for graphs, but the one that seemed most adequate was a simple line graph.

After executing the program, we are left with a new folder named JPEG with the resulting pictures.

3.2.3 Results

The results were fairly positive. For instance, in Figures 3.12, 3.13 and 3.14 we can see represented the measured values of what was meant to be a pure red color (#ff0000), pure yellow color (#ffff00) and a darker red color (#790000), respectively. As we can also see, comparing to the wave length colours that we should have (Figure 3.15), there is a blue influence in the graphics created. Considering that the measurements were taken in a black room, with minimum light interference, we can only assume that the screen itself always illuminates with some shade of blue, since other colors measured that belonged to the non-blue wavelength were also influenced.

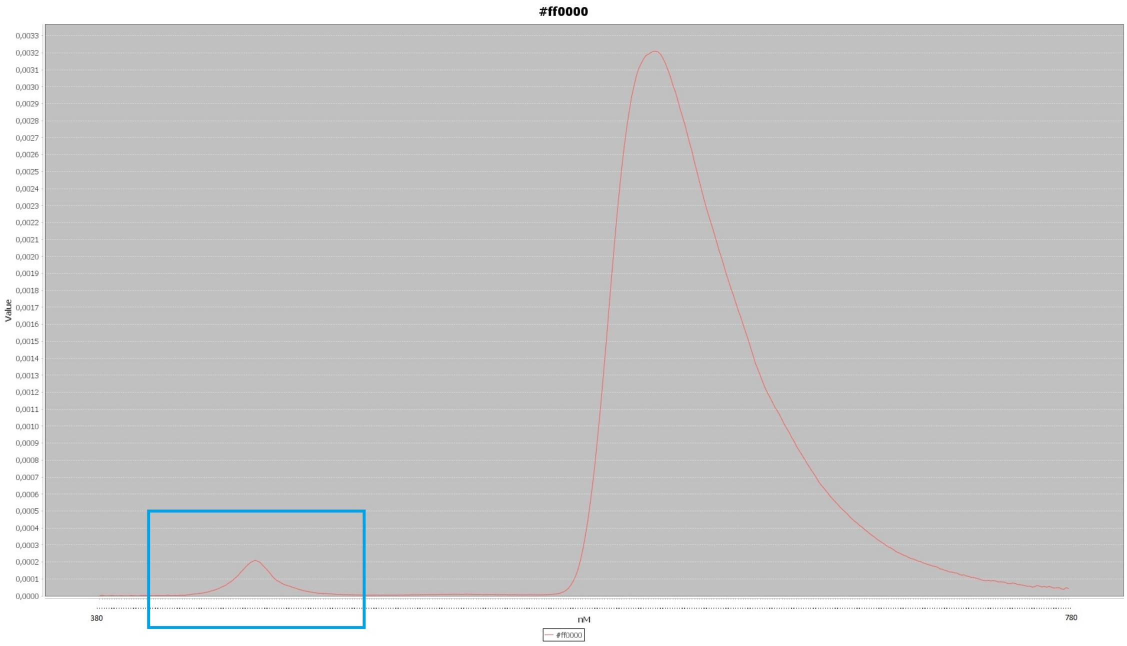

Figure 3.12: Measurements of #ff0000 with minor mistake

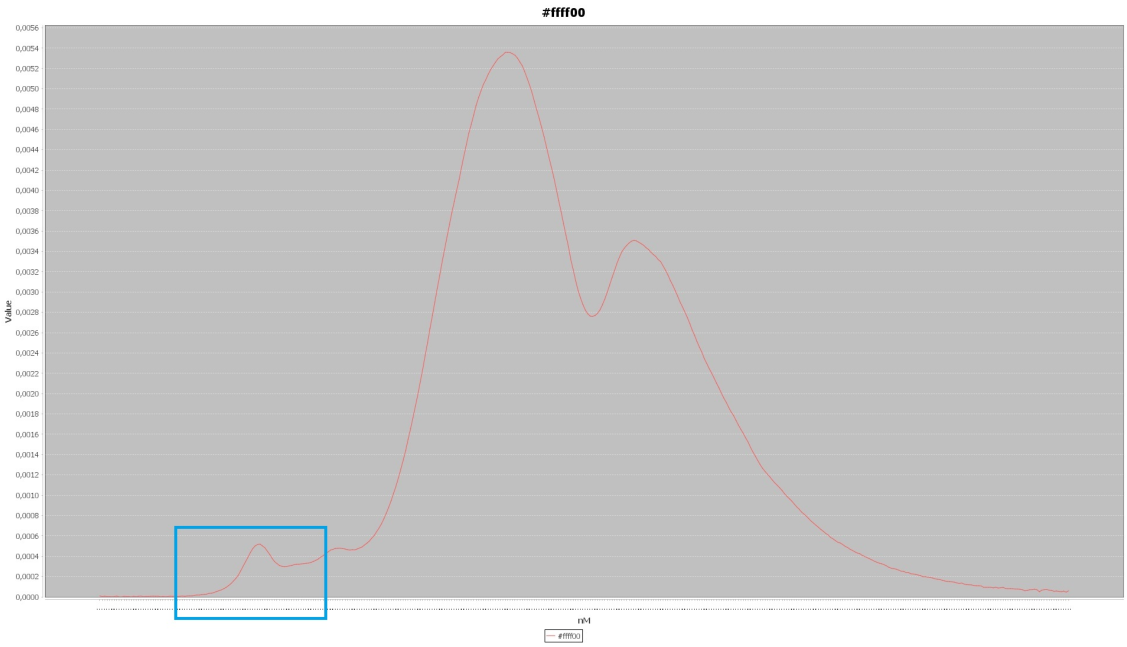

Figure 3.13: Measurements of #ffff00 with minor mistake

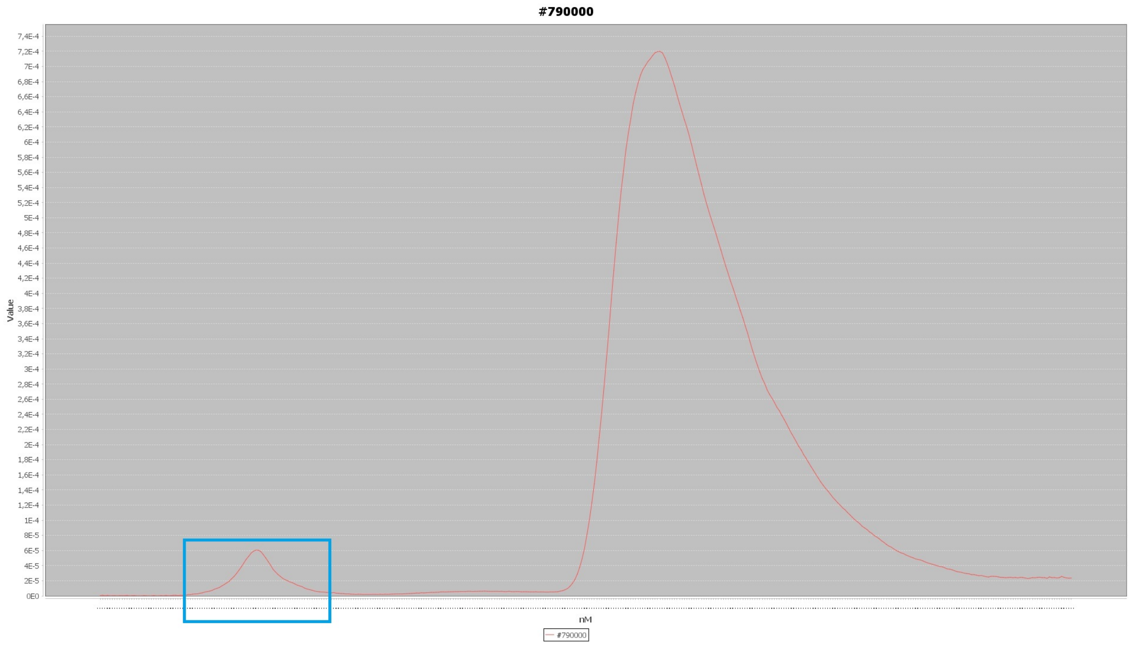

Figure 3.14: Measurements of #790000 with minor mistake

3.3 Spectral Estimation Methods

It is based on the measured data from the previous section, that the object reflectance will be calculated. There are different methods that can be applied to estimate a spectral reflectance. The three considered methods were the Wiener estimation method, which provides accurate estimates [HK07] from low-dimensional data into high-dimensional data, a linear approximation method and an Eigenvectors method.

We need to consider there are three main parts to the whole scenario (Figure 3.16): a light source, an object and a camera.

The above methods need some prior data, such as the spectral sensitivity of the camera and the spectral power distribution of the illumination. Being **L** the illuminant (smartphone screen), which was already measured, **r** the spectral reflectance of the object O being imaged, and **s** the spectral sensitivity of the camera, then the response **y** of the channel **k**, can be represented as

$$y_k = \int_{\lambda} L(\lambda) s_k(\lambda) r(\lambda) d\lambda + e_k \tag{3.1}$$

For this case, we are sampling an RGB camera, which only has three camera channels (Red, Green and Blue), meaning that it was considered that **k** = 3. e_k is the camera system noise, which was disregarded.

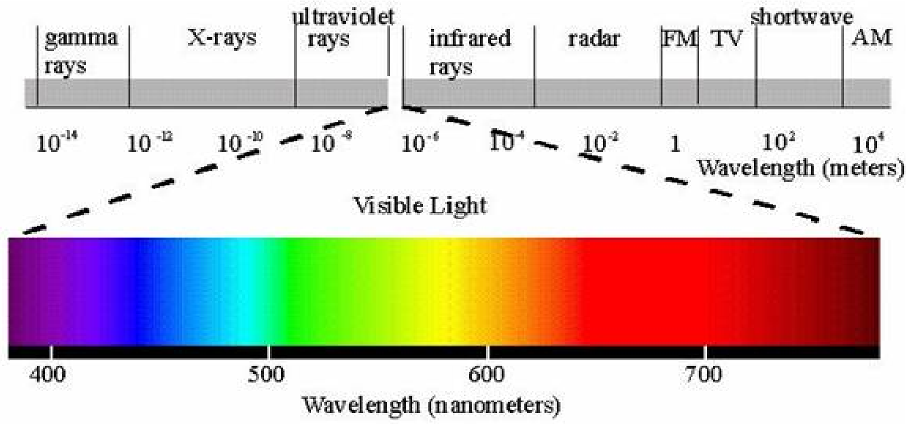

Figure 3.15: Visible spectrum with correspondent wavelength values

3.3.1 Wiener

3.3.2 Paulus

3.3.3 eigen

3.4 Spectral Metrics

To evaluate the color accuracy of the reflectance reconstruction algorithms and be able to reach a robust conclusion, the experiment was conducted using a data set with 1600 object reflectances, previously provided.

The color accuracy of the reflectance reconstruction is evaluated using both spectral and colorimetric error.

3.4.1 Root Mean Square (rms)

The spectral root mean square error is one of the methods used to compare the differences between the original reflectance $\mathbf{r}$ and its estimate $\hat{\mathbf{r}}$. It is calculated as

$$rms = \sqrt{\frac{\sum_{\lambda}[r(\lambda) - \hat{r}(\lambda)]^2}{N}} \tag{3.2}$$

where N = 401, which is the spectrum wavelength worked on (380 to 780).

3.4.2 CIELAB Color Space

Having the XYZ tristimulus values, we can calculate the colorimetric error after converting them to the CIE L*a*b* (CIELAB) color space, specified by the International Commission on Illumination (CIE), in 1976 [G.82], calculating as:

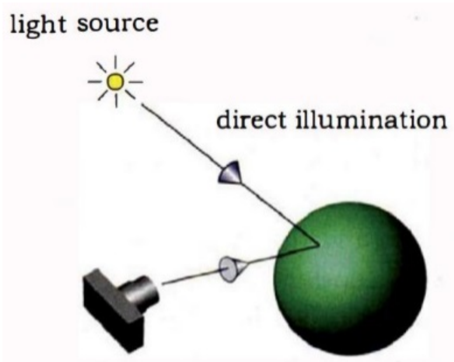

Figure 3.16: Object reflectance captured by a camera

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \tag{3.3}$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{3.4}$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{3.5}$$

where

$$f(t) = \begin{cases} \sqrt[3]{t} & ,t > 0.008856 \\ 7.787\text{t} + \frac{4}{29} & \text{, otherwise} \end{cases}$$

XYZ are the tristimulus values, which are obtained, once again, by applying the equation 3.1. X_n, Y_n and Z_n are the tristimulus values of the reference white point. The values are normalized so that Y_n = 100.

For the example of illuminant D65, that portrays standard illumination conditions at open-air, with normalization Y_n = 100, the values are

$$\text{X}_n = 95.047,$$
$$\text{Y}_n = 100.000,$$
$$\text{Z}_n = 108.883$$

Regarding the three coordinates of CIELAB,

- L* represents the lightness of the color, where L* = 0 corresponds to black and L* = 100 corresponds to white,
- a* represents the degree of red/magenta versus green (a* < 0 = green value or a* > 0 = red/magenta value) and

Table 3.1: Practical interpretation of ΔE^* color difference by Abrardo *et al.* [AVMA96]

ΔE^*	Effect
$0 > 1$	Limit of perception
$1 > 3$	Very good quality
$3 > 6$	Good quality
$6 > 10$	Sufficient
> 10	Insufficient

- b* represents the degree of yellow versus blue (b* < 0 = blue value or b* > 0 = yellow value)

When comparing two colors, such as $[L_1^*, a_1^*, b_1^*]$ and $[L_2^*, a_2^*, b_2^*]$, the CIELAB color difference is calculated as the Euclidean distance in the CIELAB space, as follows [IB99, KOO+99]:

$$\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \tag{3.6}$$

where

$$\begin{aligned}
\Delta L^* &= L_1^* - L_2^* \\
\Delta a^* &= a_1^* - a_2^* \\
\Delta b^* &= b_1^* - b_2^*
\end{aligned}$$

The XYZ values are directly related to the spectral power distribution of the colored light. What this means is that when these tristimulus values are changed, the human eye will perceive a difference in color only after the difference reaches a certain amount. The interpretation of ΔE^* can be very different, though. According to Kang [R.97], the Just Noticeable Difference (JND) is equal to 1. On the other hand, Mahy *et al.* [MVEO94] found a JND of 2.3. Two other practical interpretations can be found in the work of Abrardo *et al.* [AVMA96] and Hardeberg [Y.99] and are represented in Table 3.2 and

3.4.3 CIEDE2000 Color Difference

The CIEDE2000 definition [SD05] is considered to be the CIE recommended color difference formula, which includes new terms to improve the colorimetric error ($\Delta\, E_{00}$*) [14201, SWDA05]. The corrections added are:

- A hue rotation term, to deal with the blue region (hue angles around 275^o)

Table 3.2: Practical interpretation of ΔE^* color difference by Hardeberg [Y.99]

ΔE^*	Effect
< 3	Hardly perceptible
$3 > 6$	Perceptible, but acceptable
> 6	Not acceptable

- A compensation for neutral values
- A compensation for lightness
- A compensation for hue
- A compensation for chroma

Comparing two colors, such as [L_1^*, a_1^*, b_1^*] and [L_2^*, a_2^*, b_2^*], the modifications resulted in the equation displayed bellow:

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C}\frac{\Delta H'}{k_H S_H}^2} \tag{3.7}$$

Equation 3.7 is the one used for the colorimetric error calculation in this thesis.

3.5 Object reflectance reconstruction

To be able to reconstruct the spectral reflectance, SpectralReflectanceReconstruction.java was developed, working with MatlabControl. By executing this program, we can add different illuminant values to work with the provided camera sensitivity and the objects' spectral reflectance. Having a data set with the spectral reflectances of multiple objects (Figure 3.17) allows this study to have multiple examples, to compare the original values with the reconstructed ones, in Matlab. We can do so, using the spectral estimation methods described above.

SpectralReflectanceReconstruction.java starts off with a request for a color file, which is a required input for the execution of the script. Having found the file, it shows the colors presented in the file and opens an option menu: to add a new color measurement or to calculate with a spectral reflectance. Choosing the first one adds the measurements to the Matlab session and saves them there, for later use, if necessary.

Calculating with a spectral reflectance utilises the colors given as the illuminant, for the calculations. When selecting that option, the user can choose a specific spectral reflectance from the data set given or all of them. Choosing either case allows a new option to be selected, which defines the type of reconstruction to be used. With a single reconstruction, we obtain a graph, created by Matlab, to be able to compare the differences between the original wave and the estimated one. This option could be considered to be the most irrelevant of them all, only having usability when it proves necessary to analyse a certain reconstruction in detail. More interesting is to have several reconstructions in the same graph, to be able to see how different their estimations are, compared to the original curve. In Figure 3.18, we can observe the results for each algorithm with object 800.

For this specific object reflectance , we can conclude that the algorithm used to implement the Linear Estimation has the least amount of error compared to the original curve.

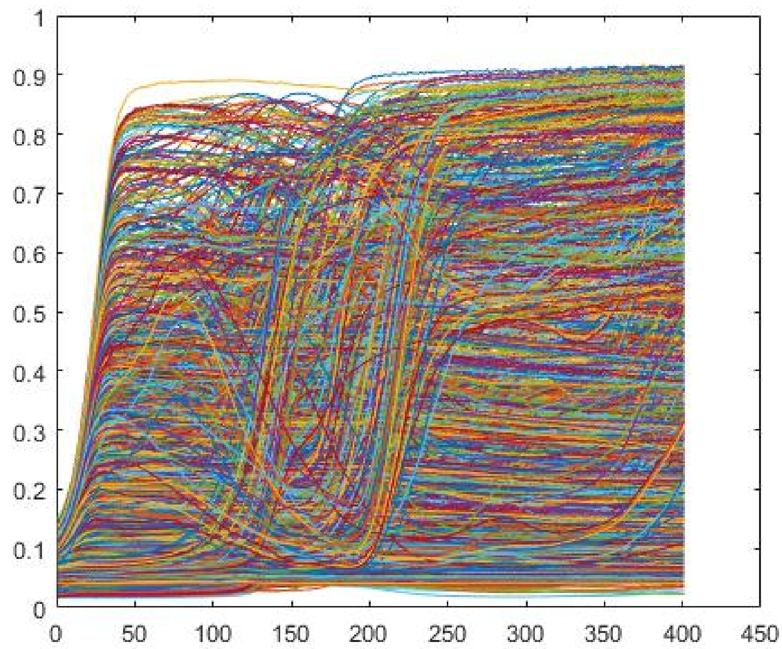

Figure 3.17: Objects' spectral reflectances with $0<\lambda<400$

In order to understand how the reconstruction works, there are some values that need to be obtained, using equation 3.1.

Having the illuminant, the sensitivity of an RGB camera and a spectral reflectance, we can calculate a response for each channel. Since there are three channels being considered and 21 colors in total in the example used, for each known spectral reflectance we will obtain 21x3 as the measurement data of the camera. Having these values, it is possible to utilise equation 3.1 once again, but having the spectral reflectance as the new data we want to obtain. Utilising the methods provided, we are able to make an estimation for each.

3.6 Experimental Results

The colors measured from the smartphone for the calibration were equidistant throughout the visible spectrum, defined every 20 nm, as shown in A.1. The reason why there's a repetition of the ff0000 color is due to the color separation margin being so small. If the colors were measured every nm, it is very likely that more colors would be repeated, due to the conversion between the visible light spectrum and its correspondent hexcode.

Having the measurements for the example file, we end up with a 401x21 matrix, representing the spectral power distribution. Utilising this, along with an RGB camera *sens* sensitivity (401x3) and a sample object's *O* spectral reflectance (401x1), then the response of each *Red*, *Green* and *Blue* channel can be calculated (equation 3.1), creating a 21x3 matrix.

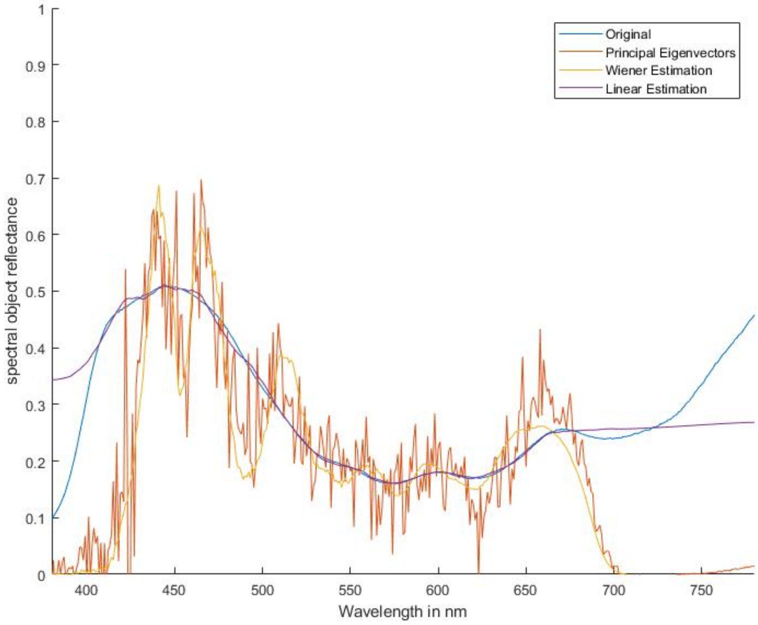

Figure 3.18: Reconstruction of the spectral object (800) reflectance using different algorithms

With the calculated data, it is possible to reconstruct the spectral reflectance of the same object, utilising the estimation methods, which results in three 401x1 matrixes that represent the Wiener, Eigenvectors and Linear estimations. We can then compare these with the original curves via rms (equation 3.2), that gives us an error value for each, for the spectral reflectance of object O.

To calculate the colorimetric error for the spectral reflectance of object O, a pair of new tristumus values are calculated (equation 3.1), one of them utilising the original wave and the other utilising the reconstructed one and both of them using a standard human sensitivity (as the *"camera"*) and CIE illuminant D65. Converting the RGB triplets to XYZ values and applying ΔE_{00}*, we obtained the colorimetric error for each estimation method.

To be able to have a robust set of results, the colorimetric and rms error were calculated for each spectral reflectance from a data set of 1600 different objects. Table 3.3 compares the spectral and the colorimetric errors of the different reconstruction methods used. A value was considered ideal, for a trustworthy reconstruction that cannot be detected by the human eye, as any value up between 0 and 1.0. Anything between 1.0 and 6.0 would still be acceptable, but given that with the Linear estimation the mean value for the colorimetric error is 0.0442 shows great results.

3.7 Summary or Conclusions

Proin vehicula pharetra urna. Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus. Ut suscipit lacus vitae dui. Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi placerat convallis.

Table 3.3: Comparison of the color accuracy of the different estimations, using the 1600 objects reflectance dataset

Method	spectral rms			ΔE_{00}* under D65		
	mean	std.	max	mean	std.	max
Wiener Estimation	0.1978	0.1244	0.4783	4.2618	2.3104	18.8325
P. Eigenvectors	0.2034	0.1284	0.5210	5.6382	3.6179	26.3720
Linear Estimation	0.0247	0.0219	0.1520	0.0442	0.0397	0.6118

Chapter 4

Conclusões e Trabalho Futuro

Deve ser apresentado um resumo do trabalho realizado e apreciada a satisfação dos objetivos do trabalho, uma lista de contribuições principais do trabalho e as direções para trabalho futuro.

A escrita deste capítulo deve ser orientada para a total compreensão do trabalho, tendo em atenção que, depois de ler o Resumo e a Introdução, a maioria dos leitores passará à leitura deste capítulo de conclusões e recomendações para trabalho futuro.

4.1 Satisfação dos Objetivos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam non felis sed odio rutrum ultrices. Donec tempor dolor. Vivamus justo neque, tempus id, ullamcorper in, pharetra non, tellus. Praesent eu orci eu dolor congue gravida. Sed eu est. Donec pulvinar, lectus et eleifend volutpat, diam sapien sollicitudin arcu, a sagittis libero neque et dolor. Nam ligula. Cras tincidunt lectus quis nunc. Cras tincidunt congue turpis. Nulla pede velit, sagittis a, faucibus vitae, porttitor nec, ante. Nulla ut arcu. Cras eu augue at ipsum feugiat hendrerit. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quam lacus, pharetra vel, aliquam vel, volutpat sed, nisl.

Nullam erat est, vehicula id, tempor non, scelerisque at, tellus. Pellentesque tincidunt, ante vehicula bibendum adipiscing, lorem augue tempor felis, in dictum massa justo sed metus. Suspendisse placerat, mi eget molestie sodales, tortor ante interdum dui, ac sagittis est pede et lacus. Duis sapien. Nam ornare turpis et magna. Etiam adipiscing adipiscing ipsum. Fusce sodales nisl a arcu. Cras massa leo, vehicula facilisis, commodo a, molestie faucibus, metus. Suspendisse potenti. Duis sagittis. Donec porta. Sed urna. Maecenas eros. Vivamus erat ligula, pharetra sit amet, bibendum et, fermentum sed, dolor. Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

4.2 Trabalho Futuro

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam tempor tristique risus. Suspendisse potenti. Fusce id eros. In eu enim. Praesent commodo leo. Nullam augue. Pellentesque tellus. Integer pulvinar purus a dui convallis consectetuer. In adipiscing, orci vitae lacinia semper, sapien elit posuere sem, ac euismod ipsum elit tempus urna. Aliquam erat volutpat. Nullam suscipit augue sed felis. Phasellus faucibus accumsan est.

Aliquam felis justo, facilisis sit amet, bibendum ut, tempus ac, dolor. Sed malesuada. Nunc non massa. In erat. Nulla facilisi. Phasellus blandit, est in accumsan cursus, libero augue elementum leo, vitae auctor mauris nisl ac tortor. Cras porttitor ornare elit. Fusce at lorem. Sed lectus tortor, vestibulum id, varius a, condimentum nec, lectus. Maecenas in nisi et magna pretium aliquam. Pellentesque justo elit, feugiat nec, tincidunt a, dignissim vel, ipsum. Sed nunc. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam tempus rhoncus leo. Donec neque quam, cursus sit amet, ultricies varius, semper non, pede. Donec porttitor. Sed aliquet feugiat elit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus tellus pede, auctor ut, tincidunt a, consectetuer in, felis. Mauris quis dolor et neque accumsan pellentesque. Donec dui magna, scelerisque mattis, sagittis nec, porta quis, nulla. Vivamus quis nisl. Etiam vitae nisl in diam vehicula viverra. Sed sollicitudin scelerisque est. Nunc dapibus. Sed urna. Nulla gravida. Praesent faucibus, risus ac lobortis dignissim, est tortor laoreet mauris, dictum pellentesque nunc orci tincidunt tellus. Nullam pulvinar, leo sed vestibulum euismod, ante ligula elementum pede, sit amet dapibus lacus tortor ac nisl. Morbi libero. Integer sed dolor ac lectus commodo iaculis. Donec ut odio.

References

[14201] CIE Publication 142. Improvement to industrial colour-difference evaluation. Vienna, Austria, Commission Internationale de L'Éclairage, 2001.

[AVMA96] Abrardo A., Cappellini V., Cappellini M., and Mecocci A. Art-works colour calibration by using the vasari scanner. pages 94–96, 1996.

[Boo00] G. Booth. *Nitro Compounds, Aromatic. Ullmann's Encyclopedia of Industrial Chemistry*. Published online, 2000.

[C.15] Katrina C. Applications of spectrophotometry in agriculture: Quantitative analysis of fertilizer properties. Available at https://www.hunterlab.com/blog/color-chemical-industry/applications-of-spectrophotometry-in-agriculture-quantitative-analysis- May 2015.

[G.82] Wyszecki G. *Color Science: concepts and methods, quantitative data and formulae*. John Wiley & Sons, 1982.

[Gil00] David Gilbert. Jfreechart. Available at http://www.jfree.org/jfreechart/, February 2000.

[Goo13] Google. Android studio. Available at https://developer.android.com/index.html, May 2013.

[Gos15] Bill; Steele Guy; Bracha Gilad; Buckley Alex Gosling, James; Joy. The java language specification. Available at https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf, February 2015.

[HK07] P. StigellK. MiyataM. Hauta-Kasari. Wiener estimation method in estimating of spectral reflectance from rgb images. 2007.

[IB99] Francisco H. Imai and Roy S. Berns. Spectral estimation using trichromatic digital cameras. 1999.

[KOO+99] Yasuhiro Komiya, Kenro Ohsawa, Yuri Ohya, Takashi Obi, Masahiro Yamaguchi, and Nagaaki Ohyama. Natural color reproduction system for telemedicine and its application to digital camera. 3:50–54, January 1999.

[Ltd14] Bentham Instruments Ltd. A guide to spectroradiometry instruments & applications for the ultraviolet. Berkshire, United Kingdom, available at https://www.bentham.co.uk, January 2014.

[Mat84] MathWorks. Matlab. Available at https://uk.mathworks.com/products/matlab.html, 1984.

[Mat10] Jeff Mather. Spectral and xyz color functions. Available at https://uk.mathworks.com/matlabcentral/fileexchange/7021-spectral-and-xyz-color-functions, Updated November 2010.

[MVEO94] M. Mahy, L. Van Eycken, and A. Oosterlinck. Evaluation of uniform color spaces developed after the adoption of cielab and cieluv. *Color Research & Application*, 19(2):105–121, 1994.

[OR06] N. Ohta and A. R. Robertson. *Colorimetry: Fundamentals and Applications*. Published in Association with the Society for Imaging Science and Technology, 2006.

[R.97] Kang H. R. *Color technology for electronic imaging devices*. SPIE Optical Engineering Press, Bellingham, Washington, USA, 1997.

[Sci15] Gamma Scientific. Gamma scientific spectroradiometers. California, United States of America, available at http://www.gamma-sci.com, May 2015.

[SD05] Wu W. Sharma, G. and E. N. Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. 2005.

[SWDA05] Gaurav Sharma, Wencheng Wu, Edul N. Dalal, and Mecocci A. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. pages 21–30, 2005.

[Wag07] Jerker Wagberg. optprop - a color properties toolbox. Available at https://uk.mathworks.com/matlabcentral/fileexchange/13788-optprop-a-color-properties-toolbox, Updated March 2007.

[Y.99] Hardeberg J. Y. *Acquisition and reproduction of colour images: colorimetric and multispectral approaches*. PhD thesis, Ecole Nationale Superieure des Telecommunications, Departament TSI, Paris, France, 1999.

Appendix A

Loren Ipsum

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado o texto usado para preencher a dissertação.

A.1 File 20colors.txt

This is an example of how a color file to be measured looks like. The colors used for the main experiment were defined as presented bellow, followed by the standard "*END*" in every color file:

#610061
#8300b5
#6a00ff
#0000ff
#007bff
#00d5ff
#00ff92
#36ff00
#81ff00
#c3ff00
#ffff00
#ffbe00
#ff7700
#ff2100
#ff0000
#ff0000
#ff0000
#D20000
#A50000
#790000

#4C0000
END

A.2 ScreenCalibration.java

A.3 De onde Vem o Loren?

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of “de Finibus Bonorum et Malorum” (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, “Lorem ipsum dolor sit amet…”, comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from “de Finibus Bonorum et Malorum” by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

A.4 Porque se usa o Loren?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using “Content here, content here”, making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for “lorem ipsum” will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

A.5 Onde se Podem Encontrar Exemplos?

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don’t look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn’t anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet.

```java
/**
 * Created by Fernandes on 07.11.2017.
 */

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.InputStreamReader;

import java.io.File;
import java.io.FileInputStream;
import java.lang.StringBuilder;

import matlabcontrol.*;

public class ScreenCalibration {

    private Socket socket;
    private PrintWriter out;
  private static PrintWriter writer;
  private static String fromServer;
  private static String fromUser;
  private static BufferedReader file;
  private static BufferedReader in;
  private static StringBuilder stringBuilder;
  private static String fileName;
  private static File measureDir;

    /**
     * Initialize connection to the phone
     *
     */
    public void initializeConnection(String filename){
        //Create socket connection
        try{
            socket = new Socket("localhost", 1234);
            out = new PrintWriter(socket.getOutputStream(), true);
      in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

      fileName = filename;

      stringBuilder = new StringBuilder();

      file = new BufferedReader(new InputStreamReader(new FileInputStream(filename)));

            // add a shutdown hook to close the socket if system crashes or exists unexpectedly
            Thread closeSocketOnShutdown = new Thread() {
                public void run() {
                    try {
                        socket.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            };

            Runtime.getRuntime().addShutdownHook(closeSocketOnShutdown);

        } catch (UnknownHostException e) {
            System.err.println("Socket connection problem (Unknown host)" + e.getStackTrace());
```

It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.