NTNU – Trondheim
Norwegian University of
Science and Technology

Development of a Smartphone-based diagnostic Tool for Jaundice

Gunnar Vartdal

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Bjørn Torger Stokke, IFY
Co-supervisor: Lise Lyngsnes Randeberg, IET
Anders Aune, St. Olavs Hospital
Tor Ramstad, IET

Norwegian University of Science and Technology
Department of Physics

Abstract

Jaundice is the cause of an estimated 114,100 deaths among newborn annually. Most of these deaths occur in the poorest regions of the world where current diagnostic technologies are too expensive. In this thesis, several methods were tested using an HTC One V and a Samsung Galaxy S3 to see whether smartphones can be used as an affordable diagnostic tool for jaundice. The methods include attaching foldable spectrometers to the phones, bandpass filters, pressing the camera lens onto the skin, and using color analysis of images taken with the phones. All methods were tested qualitatively using my own skin with either a bruise or with carrot juice applied in order to simulate an increased bilirubin concentration. The color analysis of images was also quantitatively compared to numerical simulations of skin using optical diffusion theory.

Color analysis of images taken with the phones was found to be the most promising of all the methods. The standard deviations of the color responses of both cameras were measured to be significantly smaller than the color variation of skin due to increased bilirubin concentrations. A color calibration technique intended to provide calibrations that are accurate enough for the phones to be used for bilirubin concentration measurements was developed. In its current state, it does not produce calibrations of the needed quality. A roadmap is therefore presented for the further development of the technique needed to yield satisfying results.

Sammendrag

Det er estimert at gulsott er årsaken til 114,100 årlige dødsfall blant nyfødte. De fleste av disse dødsfallene forekommer i de fattigste delene av verden hvor nåværende teknologi er for dyrt til å tas i bruk. I dette prosjektet har flere metoder blitt testet ved hjelp av en HTC One V og en Samsung Galaxy S3 for å se om smarttelefoner kan brukes som et billig diagnostisk verktøy. Brettbare spektrometere, båndpassfiltere, å trykke kameralinsen inntil huden, og fargeanalyser av bilder tatt med telefonene har blitt testet. Alle metodene ble testet kvalitativt ved å bruke min egen hud med enten et blåmerke eller med påsmurt gulrotjuice for å simulere en forhøyet bilirubinkonsentrasjon. Fargeanalysen av bilder har også blitt kvantitativt sammenlignet med numeriske simuleringer av hud utført ved hjelp av optisk diffusjonsteori.

Fargeanalyser av bilder tatt med telefonene viste seg å være den mest lovende metoden. Standardavviket til fargeresponsen til begge kameraene ble målt til å være signifikant lavere enn fargevariasjonen til hud grunnet forhøyet bilirubinkonsentrasjon. En fargekalibreringsmetode ment til å gi god nok kalibrering av kameraene til at telefonene kan bli brukt til målinger av bilirubinkonsentrasjon ble utviklet. Men kalibreringsmetoden gir per nå ikke gode nok resultater. Et veikart for den videre utviklingen av denne metoden for at den skal gi tilfredstillende resultater blir derfor presentert.

Preface

The idea of using smartphones as a cheap diagnostic tool for jaundice first came to doctor Anders Aune while he was visiting a hospital in Tanzania. He found that the hospital lacked the equipment needed to perform screening for jaundice due to the high cost of such equipment. After further research into the issue, he found that it was not a local problem occurring in Tanzania, but a major global health issue.

He therefore contacted professor of biomedical optics, Lise Randeberg, at NTNU. Together they formulated a master's thesis assignment of designing a smartphone-based diagnostic tool for jaundice. I was made aware of the thesis assignment after having contacted Engineers without borders at NTNU who put me in touch with Anders and Lise. Together we discussed and concluded that I could do this assignment as my master's thesis.

During the work with this thesis I have been participating at Applab NTNU, which is a boot camp for app developers. Since the thesis would hopefully result in a method that could be used for jaundice diagnostics, a smartphone app would have to be created implementing this method. An app prototype able to take pictures and analyze the colors of the captured image was therefore created in parallel with the development of the diagnostic methods. But due to the app not being relevant to the development of the diagnostic methods, it is not presented in this report.

Although this thesis focuses solely on developing a diagnostic tool for jaundice using smartphones, the methods developed could in theory be used to diagnose other conditions leaving some visible mark on the skin. Readers not interested in jaundice, but wanting to learn more about how smartphones can be used as cheap diagnostic tools are therefore recommended to continue reading.

During the thesis work, Lise has been my main supervisor. She has worked specifically with jaundice before and has a deep understanding and knowledge of the field of biomedical optics. This thesis could not have existed without her help. Anders has been a co-supervisor helping with questions concerning medical science. He has also been of great help, and his knowledge and position as senior consultant at St. Olavs Hospital in Trondheim will be critical in moving this project forward. I am very grateful of having been allowed to work with both of them and I look forward to our future cooperation.

Trondheim, June 2014
Gunnar Vartdal

Table of Contents

Chapter 1

Introduction

1.1 Background

Jaundice is a condition characterized by the skin of the afflicted turning yellow. This is due to elevated levels of the waste product bilirubin in the blood. The condition is therefore often called hyperbilirubinemia. It is a condition affecting approximately half of all newborn, but is in most cases harmless. The condition is still potentially dangerous because the bilirubin can accumulate in the basal ganglia of the brain, where it can cause permanent brain damage. Such brain damage, better known as kernicterus, can manifest itself as cerebral palsy, deafness, language difficulty, or in the worst cases death[1].

Bhutani et al. estimated the incidence and impairment of jaundice for 2010[2]. They concluded that failure to manage hyperbilirubinemia results in 114,100 avoidable neonatal deaths, and many grow up with disabilities. Three quarters of these deaths are estimated to occur in the poorest regions of the world in sub-Saharan Africa and south Asia. Other studies have found jaundice to be one of the top three leading causes of death among newborn in sub-Saharan Africa[1].

To prevent the usually harmless condition of jaundice from developing into kernicterus, it is highly important to identify the children at risk at an early stage. Treatment of hyperbilirubinemia is in most cases done by phototherapy, and in some extreme cases by blood transfusion. Sunlight is believed to be a cheap alternative to specialized phototherapy light-boxes, and studies are now underway investigating this[3]. It is therefore essential to be able to discover at-risk children at an early stage, so that effective treatment can be given.

Bilirubin colors the skin yellow. Jaundice can therefore often be seen visually even by people with no medical training. But mere visual judgment of the severity of jaundice has proven to be unreliable, even when performed by experienced doctors[4]. The measurement of bilirubin is therefore traditionally done by blood samples. To reduce the need of drawing blood from the newborn, devices have been developed that measure the bilirubin concentration by shining light through the

skin, so-called transcutaneous bilirubinometers[5]. Both the lab equipment needed for blood sample measurements and the devices used to measure bilirubin transcutaneously are expensive, costing more than 10,000 US dollars. Thus making them practically unavailable in low-income countries.

Since most deaths due to jaundice occur in low-income countries, there is a large unmet need of simple, reliable and affordable technologies able to identify at-risk newborn. In the last few years, cell phone technology and smartphones have penetrated into areas with little resources. This has given hope that new affordable solutions to global health issues can be developed[6].

1.2 This thesis

With this as background it should be clear that finding a way of using smartphone technology to measure the bilirubin concentration of newborn could possibly solve a major global health issue. This thesis aims at developing such a method by exploring several different ideas as to how such a measurement can be performed using smartphone cameras.

The thesis has been written for people with at least some training in physics and mathematics, either through studying physics, or having learned physics some other way, e.g. through engineering studies. More specifically, it is presumed that the reader has some background knowledge of mathematics and the electromagnetic spectrum of light. Other than that, it is hoped that the following chapter will provide the reader with enough background information to be able to follow the reasoning presented in the later chapters.

Several methods were tested over the course of this project. Some of the methods were only tested briefly and qualitatively because it was found that the methods would not likely lead to a practical measurement of bilirubin in a low-resource setting. These methods, the way they were tested, and the reasoning behind their abandonment, have still been included in the thesis. This has been done so that others wanting to develop a smartphone-based diagnostic tool for jaundice won't have to waste their time doing research that has already been done. Including these methods can also be useful in the case where the reader finds a flaw in the reasoning behind a methods abandonment. This could lead the reader to attempt to develop the method in spite of the recommendations of this thesis. Either way, including these methods could potentially be useful to the reader.

Chapter 2

Theory

The goal of this thesis was to find a way of measuring the bilirubin concentration in skin by using smartphone cameras. It is therefore useful to have knowledge of skin optics and the image creation process of modern digital cameras. This chapter serves as the background information needed to follow the reasoning presented later in the thesis, while it also includes the theoretical foundation for the numerical skin simulations and color calibrations performed during the thesis work. The following section introducing skin optics summarizes chapter two of the doctoral dissertation of Randeberg[7]. It also adds information regarding melanin and bilirubin specific to this thesis, and includes an updated value for background tissue scattering found by Bashkatov[8].

2.1 Optical properties of human skin

Skin is the human organisms barrier to the environment. It is a structure composed of different layers. The top layer, called the epidermis, is typically 100 micrometers thick and contains among other things the pigment melanin, which is the pigment responsible for the different skin colors of the world. Below the epidermis lies the dermis, which has a typical thickness of 1-4mm. In the dermis, blood vessels, connective tissue, sweat glands, hair follicles and sensory nerve systems are found. The subcutaneous fatty layer lies below the dermis, and provides insulation and protection from mechanical stress. Figure 2.1 illustrates the different skin layers along with other components found in the skin.

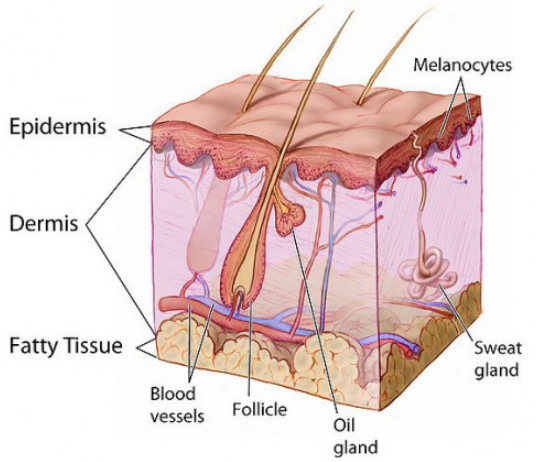


Figure 2.1: An illustration of human skin, including the skin layers and other components found in the skin. Figure taken from a web site discussing skin burns[9], distributing the figure as public domain.

Light hitting biological tissue, such as skin, is either scattered or absorbed. The intensity of light able to penetrate into the tissue is given by Beer-Lamberts law,

$$I(x) = I(0)e^{-\mu_{tr}x} \tag{2.1}$$

where $I(0)$ is the incident light intensity, x is the distance traveled in the tissue and μ_{tr} is the transport coefficient, or the total attenuation coefficient. The transport coefficient can be written as the sum of the reduced scattering coefficient, μ_s', and the absorption coefficient, μ_a.

The scattering coefficient, μ_s describes the amount of light that is scattered by the tissue. Some of this light is scattered in a forward direction, not decreasing the penetrating lights intensity. The reduced scattering coefficient incorporates this by being expressed as $\mu_s' = \mu_s(1-g)$, where g represents the amount of light scattered in a forward direction. g is called the anisotropy factor, and is calculated as the average of the cosine of the scattering angle distribution,

$$g = \overline{cos(\theta)}. \tag{2.2}$$

In skin, the anisotropy factor is approximately equal to 0.8, indicating highly forward directed scattering.

Skin contains several different molecules responsible for the absorption and scattering of incident light. The properties of these molecules and the surrounding tissue are presented in the following sections.

2.1.1 Melanin

Skin, as just mentioned, contains many absorbing and scattering molecules. The main absorber in the epidermis is melanin[10]. Skin types based on varying amounts of the pigment melanin can be classified by the Fitzpatrick skin type scale I-VI[11]. On this scale, type I refers to very fair skin that sunburns and does not tan, while type VI is at the opposite end of the scale, referring to very dark skin.

Melanin absorbs light of wavelengths ranging from ultraviolet to near-infrared. The wavelength dependence of the absorption is reported as $\lambda^{-3.46}$[12]. The absorption of melanin across the whole spectrum can therefore be defined by the absorption at a single wavelength. Absorption is often measured at 694nm, and absorption values in adults have been found to vary from $300\,\mathrm{m}^{-1}$ for fair Caucasian skin to $2500\,\mathrm{m}^{-1}$ for dark African skin[13]. Newborn skin is reported to have lower concentrations of melanin than adult skin[14]. In this thesis it is therefore assumed that the melanin absorption of newborn skin at 694nm does not exceed $2000\,\mathrm{m}^{-1}$, although exact numbers have not been found in the literature.

2.1.2 Blood

The main absorbers in blood are oxygenated and deoxygenated hemoglobin. Methemoglobin can also be formed if hemoglobin is exposed to oxidative stress, but is generally found in low concentrations. Exceptions are e.g. drug use which can lead to methemoglobinemia[15]. The absorption spectra of hemoglobin, deoxyhemoglobin and methemoglobin can be seen in Figure 2.2. The spectra of hemoglobin and deoxyhemoglobin can be seen intersecting at several points. Such points are called isosbestic points. Measuring hemoglobin concentrations is often done at isosbestic wavelengths because the total measured concentration will not depend on the oxygenation level of the blood[16].

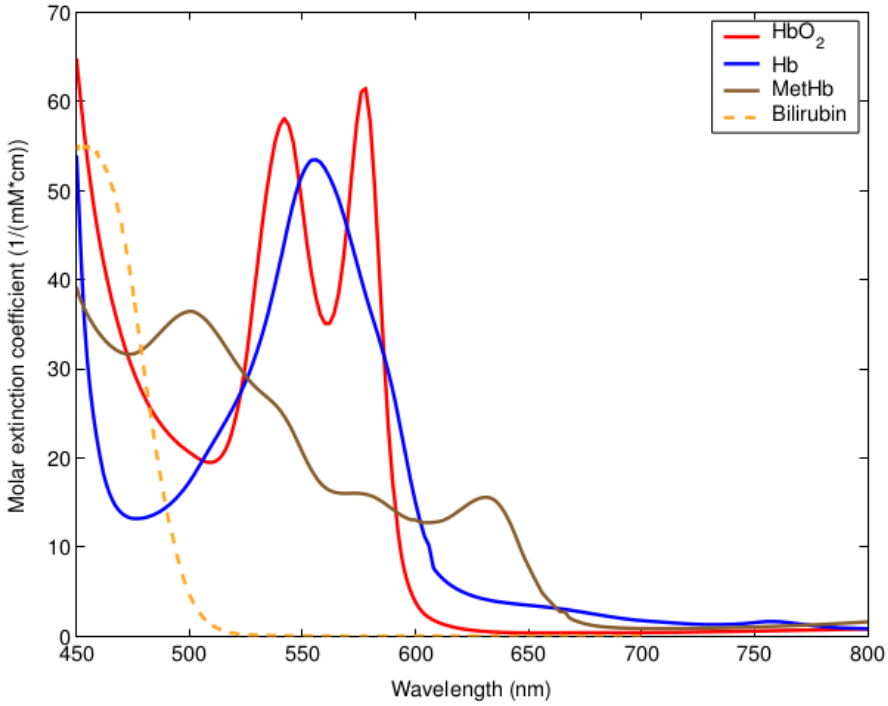


Figure 2.2: Extinction coefficients for hemoglobin, methemoglobin and bilirubin. Figure taken from Randeberg[7].

2.1.3 Bilirubin

Bilirubin is the breakdown product of heme catabolism[17]. Heme is found in hemoglobin and myoglobin. Bilirubin causes skin to turn yellow if it is allowed to accumulate in the dermis, due to its high absorption of the shorter wavelengths of the visible spectrum(see Figure 2.2). This is also the reason for the yellow color seen in old bruises[18], as macrophages are recruited to the area of the bruise where it phagocytizes erythrocytes and hemoglobin molecules, catabolizing the hemoglobin to bilirubin[19].

The yellow color from bilirubin can also be seen across the whole body, and is then caused by either a high turnover rate of hemoglobin, or liver failure, or both. Newborns acquire jaundice due to a high turnover rate of hemoglobin after birth. An elevated concentration of bilirubin in combination with a not fully developed blood-brain barrier can lead to permanent brain damage or death[20]. For this reason, 5-10% of all newborn receive either phototherapy, or in extreme cases blood transfusion to rid the body of the excess bilirubin[17].

Bilirubin in blood is bound to albumin. In this form, the combined molecules are too big to pass the blood vessels. When bilirubin concentrations exceed 400-500 micromolar, there is not enough albumin to bind all the bilirubin molecules[20]. The

free bilirubin can then diffuse through the blood vessels and into the surrounding tissue. The skin concentration of bilirubin is therefore markedly lower than the the blood serum concentration. Good correlation has been found between the skin concentration of bilirubin measured by transcutaneous bilirubinometers and the total blood serum concentration[21]. This makes it possible to estimate the blood serum concentration through transcutaneous bilirubin measurements.

Transcutaneous bilirubinometers measure the bilirubin concentration by shining light of certain wavelengths and wavelength ranges into the skin. The reflected light of each wavelength is measured and used to calculate the concentration. Full reflection spectroscopy of newborn can similarly be used to measure bilirubin concentration. In addition, the reflected spectrum allows the calculation of several other parameters such as melanin concentration and the gestational age of the newborn. For details of how such measurements are performed, the patent of a transcutaneous bilirubinometer[22] and a paper by Randeberg et al.[23] are recommended.

2.1.4 Other absorbers

Carotenoids are organic pigments found in plants. These pigments cannot be produced by animals, so they are obtained through diets. They all absorb light in the wavelength range 400-550nm. A common carotenoid abundant in carrots, betacarotene, has a double peak in its absorption spectrum at 450 and 480nm, giving it a yellow/orange color. This color, which is similar to the color of bilirubin could potentially be an error source in bilirubin measurements. But the skin concentration of all carotenoids is generally too low to have an impact[7, p. 10].

Water should also be mentioned because it is found in abundance in skin. But water has low absorption in the visible spectrum, with a minimum at 418nm and increasing absorption for wavelengths above 600nm[24]. The water content of skin is therefore not explicitly accounted for in this thesis.

2.1.5 Background tissue

Large molecules such as collagen fibers are a major source of scattering in the dermis. These molecules and changes in refraction index between them and the surrounding tissues are responsible for the fact that scattering is the dominating process in this tissue. The epidermis has similar scattering properties, but absorption due to melanin can in some cases be the dominating process in this layer. Bashkatov[8] showed that the reduced scattering coefficient of skin in the wavelength range of 400 to 2000nm can be expressed as

$$\mu_s' = 73.7\lambda^{-0.22} + 1.1 \cdot 10^{12}\lambda^{-4}. \tag{2.3}$$

Background tissue absorption is absorption caused by other molecules than the ones mentioned in the above sections. This value is set to $\mu_n = 25\,\mathrm{m}^{-1}$[25] for both the epidermis and the dermis. This value is similar to what is found in ocular(eye) tissue.

2.2 Optical diffusion

The mathematical model used for numerical simulations in this project is based on optical diffusion theory. Optical diffusion theory can be applied when scattering dominates over absorption[26]. This theory has limited validity in thin layers, and finding appropriate boundary conditions is problematic. Optical diffusion theory does not apply to air, but Haskell et al.[27] discovered boundary conditions that can be used for interfaces such as those between air and tissue, giving good results of simulations of diffuse skin reflectance[25]. Monte Carlo methods are known to be more accurate for the type of skin simulations performed in this thesis, but they are also much more computationally expensive[28] and have thus not been performed.

For the simulations performed in this project, the skin is modeled as consisting of three flat layers. The top layer represents the epidermis. To account for the papillary structure between the dermis and epidermis, as seen in Figure 2.1, blood is included in the epidermis. The epidermis therefore contains both blood and melanin in the model. The middle layer represents the top part of the dermis, and the bottom layer is a layer extending infinitely downwards. All molecules are modeled as uniformly distributed within each layer. The total transport coefficients of each layer can thus be calculated based on the background tissue scattering and absorption described in section 2.1.5, and the concentrations of the different light absorbing molecules. These transport coefficients can then be used in the diffusion model of skin developed by Svaasand et al.[25]. A summary of which will be presented below.

Svaasand et al.[25] starts by assuming an almost isotropic light distribution and by expressing the radiance L by a series expansion,

$$L = \frac{\phi}{4\pi} + \frac{3}{4\pi}\mathbf{j} \cdot \mathbf{l} + ... \tag{2.4}$$

where ϕ and $\mathbf{j}$ are the fluence rate and the diffuse photon flux vector respectively. $\mathbf{l}$ is the direction of the deviation from isotropy in the light distribution. The irradiance on a surface normal to the flux then becomes

$$E = \frac{\phi}{4} \pm \frac{j}{2}. \tag{2.5}$$

where the sign is plus for surfaces against the flux and minus for surfaces along. The diffuse photon flux vector is given by,

$$\mathbf{j} = -D\nabla\phi \tag{2.6}$$

with the diffusion constant,

$$D = \frac{1}{3\mu_{tr}}. \tag{2.7}$$

The continuity equation can then be expressed as,

$$\nabla \cdot \mathbf{j} = -\mu_a\phi + q \tag{2.8}$$

where q is the source density of diffuse photons. The combination of equations 2.6 and 2.8 yields,

$$\nabla^2\phi - \frac{\phi}{\delta^2} = -\frac{q}{D} \tag{2.9}$$

where $\delta = \sqrt{1/3\mu_{tr}\mu_a}$ is the optical penetration depth.

The boundary conditions between two scattering media is then expressed by the continuity of irradiance in the forward and backward directions,

$$\frac{\phi_1}{4} \pm \frac{j_1}{2} = \frac{\phi_2}{4} \pm \frac{j_2}{2}. \tag{2.10}$$

Haskell et al. found a very useful boundary condition at the skin-air interface by relating the reflected part of the irradiation at the inside of the interface to the irradiation propagating back into the skin[27]

$$R_{eff}(\frac{\phi}{4} + \frac{j}{2}) = \frac{\phi}{4} - \frac{j}{2} \tag{2.11}$$

where R_{eff} is the effective reflection coefficient. The value of R_{eff} can be found by integrating the Fresnel reflection coefficient for unpolarized light over all angles of incidence.

For an isotropic light distribution, the source density functions of Equation 2.9 are expressed as functions of the light intensity, P_0, transmitted through the skin-air interface as

$$\begin{aligned} q_1 &= P_0\mu'_{s,1}e^{-\mu_{tr,1}x} \\ q_2 &= P_0\mu'_{s,2}e^{-\mu_{tr,1}d_1}e^{-\mu_{tr,2}(x-d_1)} \\ q_3 &= P_0\mu'_{s,3}e^{-\mu_{tr,1}d_1}e^{-\mu_{tr,2}d_2}e^{-\mu_{tr,3}(x-d_1-d_2)} \end{aligned} \tag{2.12}$$

where the indices 1,2 and 3 represents each layer, d represents the thickness of a layer, and x the distance from the skin surface.

The solutions to equation 2.9 using these source equations can then be written

$$\begin{aligned} \phi_1 &= \frac{P_0\delta_1^2\mu'_{s,1}}{D_1(1-\mu_{tr,1}^2\delta_1^2)}e^{-\mu_{tr,1}x} + A_1e^{-\frac{x}{\delta_1}} + A_2e^{\frac{x}{\delta_1}} \\ \phi_2 &= \frac{P_0\delta_2^2\mu'_{s,2}}{D_2(1-\mu_{tr,2}^2\delta_2^2)}e^{-\mu_{tr,1}d_1}e^{-\mu_{tr,2}(x-d_1)} + A_3e^{-\frac{x}{\delta_2}} + A_4e^{\frac{x}{\delta_2}} \\ \phi_3 &= \frac{P_0\delta_3^2\mu'_{s,3}}{D_3(1-\mu_{tr,3}^2\delta_3^2)}e^{-\mu_{tr,1}d_1}e^{-\mu_{tr,2}d_2}e^{-\mu_{tr,3}(x-d_1-d_2)} + A_5e^{-\frac{x}{\delta_3}} \end{aligned} \tag{2.13}$$

The values of the constants $A_1 - A_5$ can then be found by applying the boundary conditions of equation 2.10 and 2.11. After this, the diffuse reflection coefficient can be calculated by

$$\gamma = \frac{j|_{x=0}}{P_0}. \tag{2.14}$$

For the complete expression for γ, the reader is referred to the appendix of Svaasand et al.[25].

2.3 Color spaces

The eye has three types of cone cells which are sensitive to light of varying wavelengths. These cone cells provide the sensory input needed for color perception. One type of cone cells primarily absorbs light of shorter, blue, wavelengths, and the other two absorb mainly green and mainly red, respectively. Although there is significant overlap between the sensitivity spectra of the cells. Three parameters corresponding to the stimulus values provided by the cone cells can therefore be used to describe any perceivable color.

2.3.1 XYZ

The CIE XYZ color space is one such tristimulus color space representation. It was created by the International Commission on Illumination(CIE) in 1931[29]. The X, Y and Z values are calculated by the following integrals

$$\begin{aligned} X &= \int_{380}^{780} I(\lambda)\bar{x}(\lambda)d\lambda \\ Y &= \int_{380}^{780} I(\lambda)\bar{y}(\lambda)d\lambda \\ Z &= \int_{380}^{780} I(\lambda)\bar{z}(\lambda)d\lambda \end{aligned} \tag{2.15}$$

where $I(\lambda)$ is the spectral distribution of the light, and $\bar{x}$, $\bar{y}$ and $\bar{z}$ are color matching functions. For light reflected off a surface, the spectral distribution can be expressed as

$$I(\lambda) = P(\lambda)R(\lambda) \tag{2.16}$$

where $P(\lambda)$ is the spectral distribution of the light source, and $R(\lambda)$ is the reflection coefficient of the surface.

The color matching functions $\bar{x}$, $\bar{y}$ and $\bar{z}$ have been constructed to be similar to the sensitivities of the cone cells in the eye. Since cone cells are not distributed equally within the eye, the CIE uses the response of the cone cells located in the center 2 degrees of the field of vision when creating the color matching functions. These color matching functions are therefore called the CIE 2^o standard observer, and can be downloaded from the CIE websites[30]. Other standard observers have been created similarly, such as the CIE 10^o standard observer, with similar color matching functions, $\bar{x}_{10}$, $\bar{y}_{10}$ and $\bar{z}_{10}$.

The color matching function $\bar{y}$ was designed to match the perceived brightness or luminance of a color. This could be done because humans perceive the brightness or luminance of a color by mostly using the intensity of the green light in the spectrum. By slightly deviating from the measured spectral sensitivity of the cone cells sensitive to the center wavelengths, $\bar{y}$ could be designed to match this perceived brightness. This means that the Y value of the XYZ color space is used as a measure of how bright a color is. By normalizing the XYZ values, a color can be represented

by two chromaticity coordinates and a separate brightness or luminance value. If the XYZ values are normalized as

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} = 1-x-y \end{aligned} \tag{2.17}$$

the colors can be fully represented by the x and y chromaticity coordinates and the Y luminance coordinate. This is called the xyY color space.

2.3.2 sRGB

Another useful color space is the sRGB color space. sRGB has become the industry standard color space for LCDs, digital cameras, printers, scanners, and of course smartphones. It is therefore important to be able to convert sRGB colors to XYZ and xyY and vice versa if analysis using these color spaces are of interest. Conversion from XYZ to sRGB is done by first finding linear RGB values.

$$\begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.18}$$

These linear RGB values can then be converted to sRGB values by

$$C_{srgb} = \begin{cases} 12.92C_{linear}, & C_{linear} \leq 0.0031308 \\ 1.055C_{linear}^{\frac{1}{2.4}} - 0.055, & C_{linear} > 0.0031308 \end{cases} \tag{2.19}$$

where C is replaced by R, G, or B. These values for RGB will be in the range 0 to 1 if the XYZ values are similarly normalized. If values from 0 to 255 are needed, as is common for displays or image values, the usual technique is to multiply by 255 and round to an integer.

The reverse transformation is performed by first calculating the linear RGB values

$$C_{linear} = \begin{cases} \frac{C_{srgb}}{12.92}, & C_{srgb} \leq 0.04045 \\ (\frac{C_{srgb}+0.055}{1.055})^{2.4}, & C_{srgb} > 0.04045 \end{cases} \tag{2.20}$$

and then calculating the matrix product

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix}. \tag{2.21}$$

2.3.3 CMYK

The CMYK color space is another color space worth mentioning. The primary colors of this color space is cyan, magenta and yellow. Hence the CMY. The K stands for key, and represents black. This convention is useful in printers because black ink can then be used directly in print to darken the image. Something that can be both hard and expensive to do using the ink of the three other colors.

2.4 Image creation in smartphone cameras

Almost all smartphone cameras today include a CMOS sensor. CMOS is an abbreviation for Complimentary Metal-Oxide Semiconductor. These sensors have arrays of photodiodes that generate current when photons are absorbed. The efficiency with which the photodiodes generate current depends on the wavelength and is called the quantum efficiency. A typical quantum efficiency of a CMOS sensor can be seen as the black line in Figure 2.3.

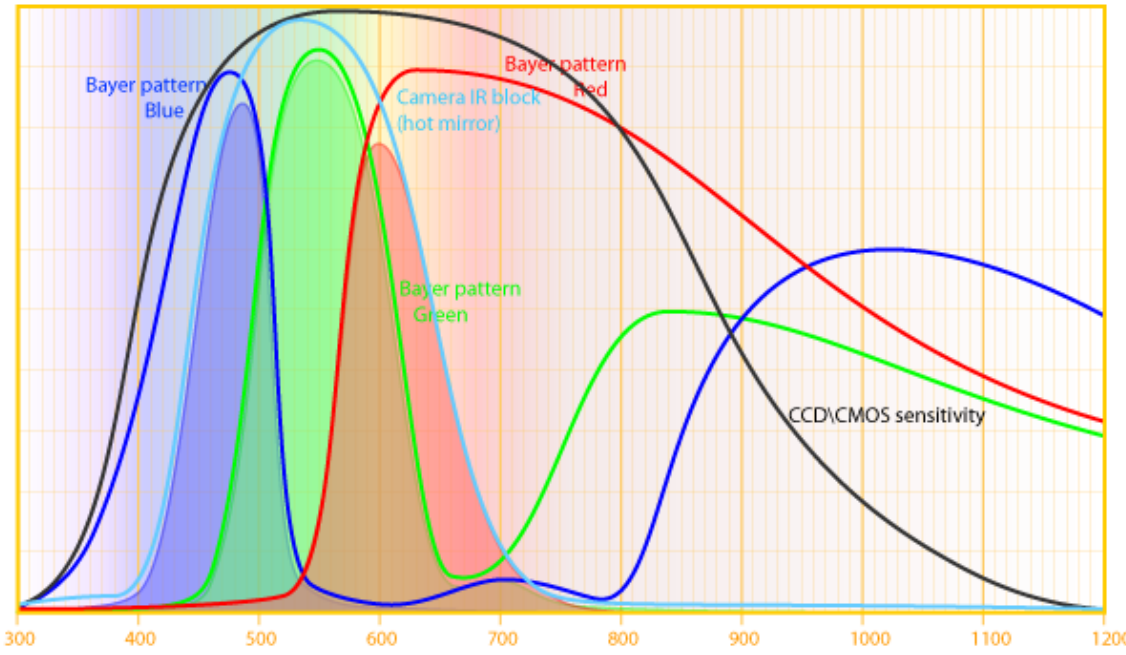


Figure 2.3: Figure showing the quantum efficiency of the CMOS sensor(black line), the transmission of the cameras infrared filter(light blue line), and the transmission of the red, green and blue wavelengths through a Bayer color filter array. Figure taken from ir-photo.net[31], which distributed it under a Creative Commons license[32].

On top of the CMOS sensor is a color filter array. A single cell in a color

filter array covers a single photodiode and transmits only wavelengths of a certain color to this diode. A commonly used filter of this type is the Bayer filter array. The Bayer filter array contains 50% green, 25% red and 25% blue filters. This is to resemble the human eye's increased sensitivity to the intensity of green light. The filters are placed in a specific pattern as can be seen in Figure 2.4, and their transmission frequencies can be seen in Figure 2.3.

This type of filter arrangement creates in effect three different images, one red, one green and one blue. All three of these images are collectively called a raw format image. High end digital cameras have the option to output images in this format. Cheaper cameras, including smartphone cameras, do not have this option. In these cameras the three original images are combined using a demosaicing algorithm, interpolating the missing red, green or blue pixel values from the surrounding pixels. This process can be done by both hardware and software, and the specific algorithms used differs for the different cameras. The result of this process in smartphones is a JPEG image with RGB values for each pixel in the sRGB color space.

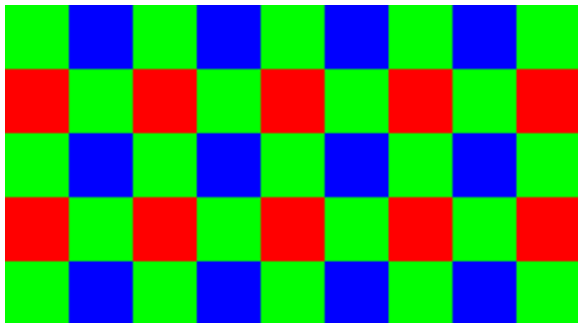


Figure 2.4: Illustration of the positions of the red, green and blue color filters in a Bayer filter array.

White balance adjustments of the image are also performed in addition to the demosaicing algorithm. These color adjustments are performed to attempt to recreate the colors of the scene more accurately. The process is called white balance because photographers often use images of known white or gray objects as references when performing these adjustments. The images need such color adjustments because the light source illuminating the scene will create different color responses in the camera depending on the light source used. Cameras therefore often come preset with white balance settings such as daylight, cloudy, incandescent and fluorescent to accommodate for common light sources. The cameras also have an auto white balance setting which adjusts the colors of the images automatically depending on the distribution of colors in the image.

The white balance mode can be set using the smartphone camera app on almost all smartphones. Other settings, such as the shutter speed, which is the amount of time the camera allows light to reach the sensor, can only be set on a few smartphones. This lack of control over camera settings could pose a challenge

because small changes in e.g. the light intensity of a scene could potentially alter the image substantially.

2.5 Color calibration

White balance adjustments can create images that look good, but that does not mean that colors are reproduced accurately. To achieve good color reproduction, images of objects with known colors can be captured. The colors of the captured images can then be adjusted according to the known colors of this color target. A common target used in photography settings is the Macbeth ColorChecker[33]. The ColorChecker contains 24 squares of different colors. The upper half contains colors often found in nature, while the bottom half contains a gray scale and colors close to the primary colors of the RGB and CMY color spaces. An image of the ColorChecker used in this project can be seen in Figure 2.5.

Two different calibration methods were tested in this project. Both of which are presented in the following sections.

Figure 2.5: Image of the Macbeth ColorChecker used in this project. This image was taken using an HTC One V with the fluorescent white balance setting. The light source was a combination of fluorescent light bulbs and daylight.

2.5.1 General Polynomial Transform

Ilie et al.[34] tested three different methods for the color calibration of images. The first was a linear least squares matching, the second was a linear RGB to RGB matrix transformation, and the third was a general polynomial transform. Of these, the general polynomial transform was found to be the most accurate because it was the only method that could account for both linear and non-linear error sources. It was therefore chosen to be implemented in this thesis.

The equation for the general polynomial transform for color channel $c \in \{r, g, b\}$ of sample color s is

$$\sum_{k=1}^{D}(t_{rc_k}Ir_s^k + t_{gc_k}Ig_s^k + t_{bc_k}Ib_s^k) + t_{c_0} \simeq Tc_s \tag{2.22}$$

where D is the degree of the polynomial approximation. Ir_s^k, Ig_s^k and Ib_s^k are the red, green and blue sample color values of the captured image raised to the power of k, while Tc_s is the true value of the target color sample s. t_{xc_k} is the polynomial coefficient of k^{th} order, specifying the influence of the input color channel $x \in \{r, g, b\}$ on the output color channel c. For $D = 2$ with 24 color samples, such as when using the ColorChecker, equation 2.22 can be written in matrix form as

$$\begin{bmatrix} Ir_1 & Ir_1^2 & Ig_1 & Ig_1^2 & Ib_1 & Ib_1^2 & 1 \\ Ir_2 & Ir_2^2 & Ig_2 & Ig_2^2 & Ib_2 & Ib_2^2 & 1 \\ ... & ... & ... & ... & ... & ... & ... \\ Ir_{24} & Ir_{24}^2 & Ig_{24} & Ig_{24}^2 & Ib_{24} & Ib_{24}^2 & 1 \end{bmatrix} \times \begin{bmatrix} t_{rc_1} \\ t_{rc_2} \\ t_{gc_1} \\ t_{gc_2} \\ t_{bc_1} \\ t_{bc_2} \\ t_{c_0} \end{bmatrix} \simeq \vec{Tc}_s \tag{2.23}$$

This equation can be solved for the polynomial coefficients, $\vec{t}_{c_k}$, by calculating the pseudo-inverse of the matrix, B, containing the input sample color values. The equation to be solved is therefore

$$B \times \vec{t}_{c_k} \simeq \vec{Tc}_s \leftrightarrow \vec{t}_{c_k} \simeq Pinv(B) \times \vec{Tc}_s \tag{2.24}$$

resulting in a vector $\vec{t}_{c_k}$ that can be used to convert any input color from the input color space to the calibrated color space. The equations outlined here assumed usage of an RGB color space, but the method can be used for any three-dimensional vector space, including the XYZ color space.

2.5.2 Thin-Plate Spline Interpolation

Menesatti et al.[35] compared a commonly used commercial color profiling tool called ProfileMaker to a novel calibration procedure using thin-plate spline interpolation. The thin-plate spline method was found to give significantly better calibration results than the commercial profiling tool.

The thin-plate spline interpolation method is named after the physical analogy of bending thin metal plates to fit to certain fixed coordinates[36]. It is used in the field of medical imaging, as a means of transforming and analyzing images from e.g. magnetic resonance imaging(MRI) scans[37]. In three dimensions the method works by finding a function $f(x_1, x_2, x_3)$ that minimizes

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_1(i), x_2(i), x_3(i)))^2 + \lambda J(f) \tag{2.25}$$

where n is the number of known reference sample points, y_i is a value at such a sample point, $x_1(i)$, $x_2(i)$ and $x_3(i)$ are the input coordinate values of sample i, and λ is a smoothing parameter determining the effect the penalty function $J(f)$ will have on the final interpolation. $J(f)$ represents the bending energy of the thin

plates. In three dimensions this function is given by[38, p. 89]

$$J(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\frac{\partial^2 f}{\partial x_1^2})^2 + (\frac{\partial^2 f}{\partial x_2^2})^2 + (\frac{\partial^2 f}{\partial x_3^2})^2 + 2[(\frac{\partial^2 f}{\partial x_1 x_2})^2 + (\frac{\partial^2 f}{\partial x_1 x_3})^2 + (\frac{\partial^2 f}{\partial x_2 x_3})^2]dxdydz. \tag{2.26}$$

Duchon[39] showed that the interpolation function f minimizing 2.25 is of the form

$$f(\mathbf{x}) = a_1 + a_2 x_1 + a_3 x_2 + a_4 x_3 + \sum_{i=1}^{n} b_i U(|\mathbf{x} - \mathbf{x}(i)|) \tag{2.27}$$

where $U(r) = r^2 \log(r^2)$.

By defining a matrix K with elements $K_{ij} = U(|\mathbf{x}(i) - \mathbf{x}(j)|)$ and a matrix $M = K + n\lambda I$, where I is the identity matrix, Wahba[40] shows that the equations above can be written as

$$\begin{aligned} M\mathbf{b} + T\mathbf{a} &= \mathbf{y} \\ T'\mathbf{b} &= 0 \end{aligned} \tag{2.28}$$

where $\mathbf{a}$ and $\mathbf{b}$ are vectors containing the coefficients in equation 2.27, $\mathbf{y}$ is a vector with values of the n reference samples, and T is a matrix of the form

$$T = \begin{bmatrix} 1 & x_1(1) & x_2(1) & x_3(1) \\ 1 & x_1(2) & x_2(2) & x_3(2) \\ ... & ... & ... & ... \\ 1 & x_1(n) & x_2(n) & x_3(n) \end{bmatrix} \tag{2.29}$$

Equation 2.28 can thus be expressed in matrix form as

$$\begin{bmatrix} M & T \\ T' & O(4,4) \end{bmatrix} \times \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ O(4,1) \end{bmatrix} \tag{2.30}$$

where $O(r, c)$ is a zero matrix. The values for the coefficients in vectors $\mathbf{a}$ and $\mathbf{b}$ are then calculated by inversing the matrix $\begin{bmatrix} M & T \\ T' & O(4,4) \end{bmatrix}$. To include the calibration of all three dimensions in the above equation, vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{y}$ can be extended to matrices with three columns of the form $\begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix}$.

For a more rigorous mathematical development of the above relations, the reader is referred to chapter 2.4 of the book by Wahba[40].

Chapter 3

Materials and Methods

A Samsung Galaxy S3(GT-I9300, Samsung, South-Korea) and an HTC One V(One V, HTC, Taiwan) were used for this project. Compared to other smartphones, the S3 has a high quality camera[41], while the One V has a decent camera[42]. Both cameras were bought used. The S3 was bought through a private person and the One V was bought through Green Phone Security AS, a company that specializes in reselling used mobile phones.

Several methods were tested using these two phones. They were tested on my own skin which is of a light color, type II on the Fitzpatrick skin type scale. For control purposes, numerical skin simulations and measurements of my skin with bruises and carrot juice were performed. The methods tested and the controls used are described in the following sections.

3.1 Controls

In order to test whether the different measurement methods could separate regular skin from skin with higher bilirubin concentrations, skin with higher bilirubin concentrations was needed. Testing on newborn babies with corresponding bilirubin blood serum measurements was the obvious choice, but acquiring ethical approval of such tests take a long time. In addition, the description of the measurements in the application for ethics approval need to be very detailed, leaving little room for adjustments of the methods. In a beginning research stage, it would therefore be useful to perform coarse tests of the methods before developing more refined protocols.

3.1.1 Carrot juice and bruises

There exists a couple of methods to either simulate increased bilirubin concentration in the skin or to obtain an actual higher skin bilirubin concentration. One of these methods is to perform measurements on bruised skin. The yellow color seen

in or around bruises is, as mentioned in section 2.1.3, caused by increased bilirubin concentration. Measurements on both yellow bruised skin and skin in close proximity to the bruise, but with no distinct yellow color, have been performed. This gives a qualitative result of whether the measurement method can separate skin with low bilirubin concentration from skin with higher concentrations.

Carrot juice can also be applied to the skin in order to simulate an increased bilirubin concentration. A comparison of the absorption spectrum of bilirubin and an attenuation spectrum of carrot juice applied to my arm can be seen in Figure 3.1. The shape of the spectra are quite similar, but the spectrum of carrot juice is shifted approximately 50 nanometers to the right of that of bilirubin. This is not ideal, but in lack of better alternatives, such as bruises, carrot juice was used instead. The carrot juice was applied by pouring it onto the skin generously. It was then allowed to dry in the air before measurements were performed.

The carrot juice(organic carrot juice, Biotta AG, Switzerland) attenuation spectrum seen in Figure 3.1 was created using reflection spectra measured by a reflectance integrating sphere(ISP-REF, Ocean Optics, Netherlands) connected to an SD2000 spectrometer(SD2000, Ocean Optics, Netherlands) with the SpectraSuite(SpectraSuite, Ocean Optics, Holland) computer application. A 10 millisecond integration time was used along with smoothing by using the average of ten spectra as the resulting spectrum. First, the reflection spectrum of the skin of my forearm was measured by placing the integrating sphere on my forearm with light pressure. Then the carrot juice was applied to the same area of my forearm. After the carrot juice had dried, a second reflection spectrum was measured at the same spot as the first one, with similar light pressure. The attenuation spectrum of the carrot juice is therefore the reflection spectrum of my forearm with carrot juice subtracted from the reflection spectrum of my forearm without the carrot juice.

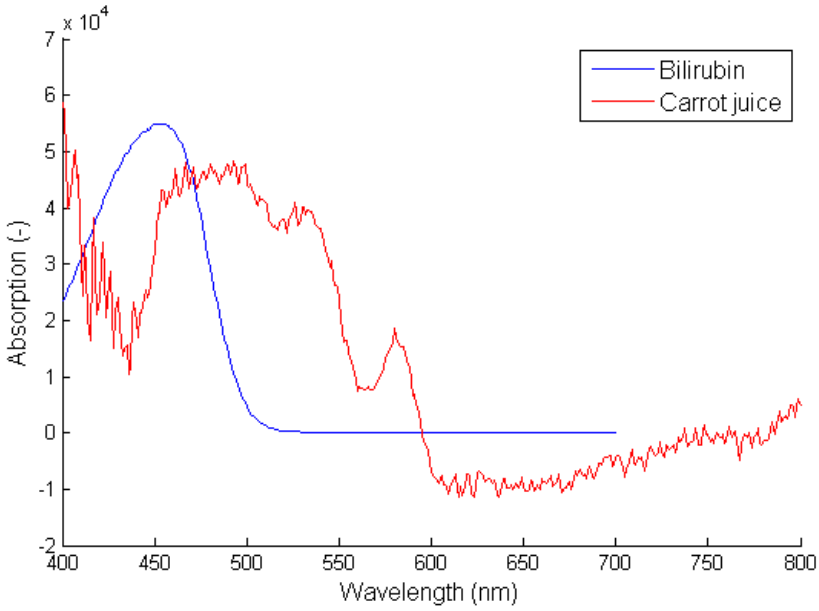


Figure 3.1: The attenuation spectrum of carrot juice applied to my arm is shown in red. The absorption spectrum of bilirubin is shown in blue. The attenuation spectrum of carrot juice has been scaled to approximately the same level as bilirubin for easier comparison. Bilirubin absorption data was taken from the Oregon Medical Laser Center[43].

3.1.2 Numerical simulation of skin

The use of carrot juice and bruises allowed for a coarse test of whether the different measurement methods would work or not. But quantitative results were also needed to test the accuracy of the promising methods. Therefore, numerical simulations of skin were performed. This allowed for the creation of both reflection spectra and simulated colors of skin with varying concentrations of e.g. bilirubin and melanin.

Simulations were performed using the three-layer diffusion model described in section 2.2. Wavelengths were sampled at 5nm intervals. The matlab scripts used are given in appendix A.2. The resulting simulated reflection spectra were then combined with a light source spectrum similar to one used during measurements, i.e. CIE standard illuminant D50 for daylight[44, p. 93], through equation 2.16 and converted to a color using equation 2.15.

Unless otherwise specified, the simulations were performed with a blood oxygenation level of the top and second layers of 0.5 and 0.8 respectively. The blood volume fractions of the top and second layer were set to 1%. The thickness of the top layer was set to 100 microns, while the thickness of the second layer was set to 250 microns. The water, fat, betacarotene and methemoglobin levels were set to zero. The scattering coefficient was calculated with values as described in

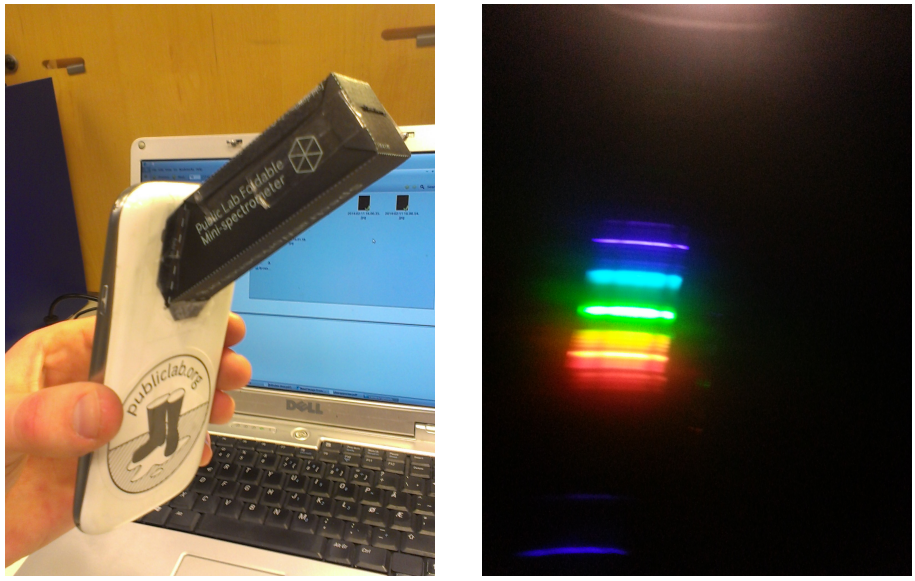
equation 2.3. Melanin concentrations were varied from an absorption at 694nm of $250\,\mathrm{m}^{-1}$ to $2000\,\mathrm{m}^{-1}$. Bilirubin concentrations were varied from 0 micromolar to 200 micromolar.

3.2 Methods

3.2.1 Foldable spectrometer

Spectroscopy of newborn jaundiced skin allows for the calculation of the bilirubin concentration as well as other skin properties, as mentioned in section 2.1.3. It would therefore be useful to transform the camera of the smartphones into spectrometers. Two foldable spectrometers(Foldable Mini-Spectrometer, Public Lab, USA) designed specifically to be attached to cameras were ordered through publiclab[45]. This type of spectrometer was chosen since it is both cheap(10 USD) and easy to assemble[46] while using common materials such as cardboard and cd-roms. These two properties will in theory make it easier to replace and repair broken equipment in low-resource settings.

The slit opening of the spectrometers were measured to be $0.7 \pm 0.1\,\mathrm{mm}$. The slit opening of one of the spectrometers was increased to $2 \pm 0.5\,\mathrm{mm}$ using scissors. It was not possible to cut perfectly straight, hence the increased error in the slit opening size. The foldable spectrometers were attached to the phones using scotch tape. Measurements were performed by setting the white balance mode of the camera to something other than auto to prevent the smartphones automatic color adjustments. Then the spectrometer was pointed at the object of interest before an image was taken in a regular way using the built-in camera app of the phone. An image of the setup along with a resulting image using the spectrometer can be seen in Figure 3.2.



(a) **(b)**

Figure 3.2: (a) Image showing one of the foldable spectrometers attached to the Samsung Galaxy S3. (b) Image taken while pointing the spectrometer at a fluorescent light bulb.

The image taken by the smartphone camera with the foldable spectrometer attached, such as the image in Figure 3.2b, was converted to a spectrum by summing the pixel color values either horizontally or vertically depending on the orientation of the image. Matlab scripts able to do this were found free online[47]. But since not all images captured using the spectrometers resulted in colored lines that were parallel to the horizontal or vertical axis, as in Figure 3.2b, the images were rotated before a spectrum was created. This was done using the image manipulation software GIMP(GNU Image Manipulation Program, GNU).

Color images contain red, green and blue channels. A spectrum can be created for each channel, but the sum of the values of all three channels can also create a spectrum. By plotting the spectrum of each color channel, the separation of noise from spectral data became an easier task. An example of this can be seen in Figure 4.1, where there is contribution to the spectrum from the blue and green channels at wavelengths above 600 nanometers, and also contribution to the spectrum from the red channel at wavelengths below 450 nanometers.

Wavelength calibration of the spectrometers was done by measuring the spectrum of a fluorescent light bulb using the spectrometers. Fluorescent light bulbs contain mercury which produce several characteristic peaks in the resulting light spectrum. These peaks can be seen as labeled peaks 1, 2, 5 and 7 in Figure 3.3. The foldable spectrometers do not provide the high resolution needed to be able to discern the minor peaks 1 and 7 in the resulting spectra. Therefore, only peaks

2 and 5 were used for calibration. Using only two peaks limits the calibration to be a linear function of the pixel positions in the image. It is likely that this limits the wavelength calibration to be accurate only around these two peaks, which have wavelengths of 435.8 nm and 546.1 nm. The script found in appendix A.1 was used for the calibration. This calibration procedure had to be done each time the spectrometers were attached to the phones, since a slight misplacement of the spectrometer could lead to a relatively large positional and rotational change in the resulting image.

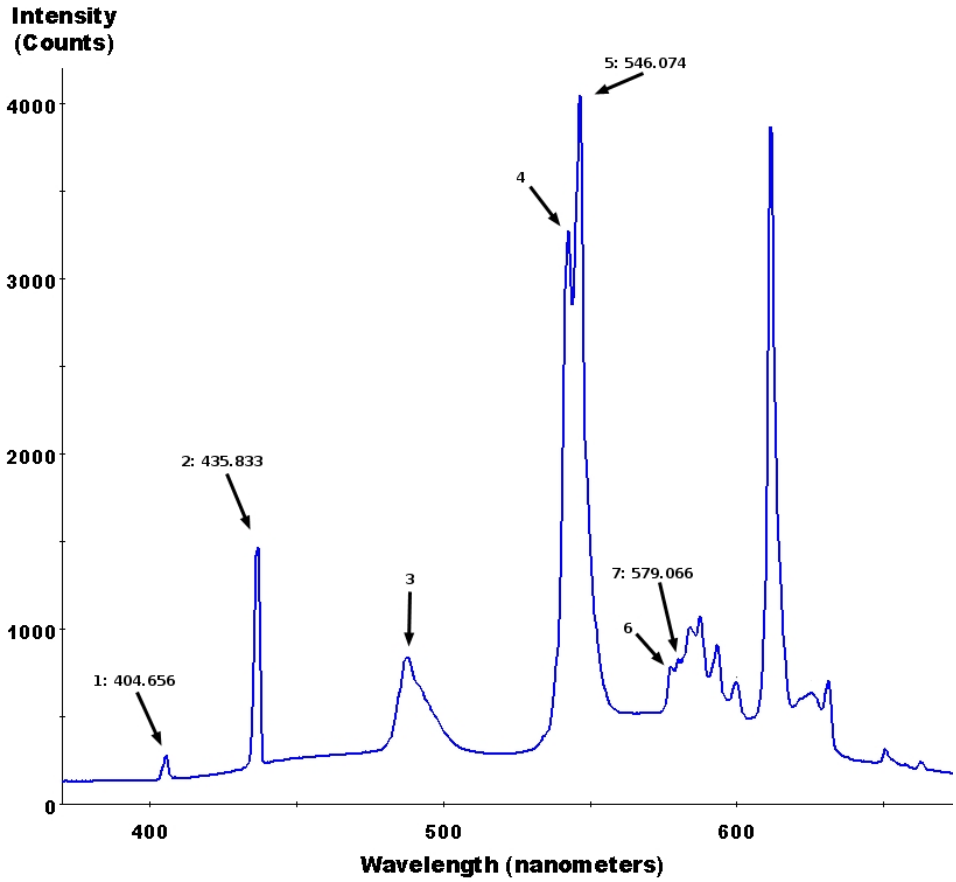


Figure 3.3: A typical spectrum of a fluorescent light bulb with labeled peaks. Peaks 1, 2, 5 and 7 are due to the mercury found in all fluorescent light bulbs. This figure was taken from Wikimedia Commons[48] and slightly modified by adding the wavelengths of the mercury peaks to the peak labels.

The spectra measured by pointing the spectrometers at objects other than the light source are reflection spectra. To convert these spectra to attenuation spectra,

the reflection spectrum of a white object, such as a white sheet of paper, was measured. The reflection spectrum of e.g. skin was then subtracted from this white reference spectrum to obtain the attenuation spectrum of the skin. A similar approach was used to measure the attenuation due solely to carrot juice. The only difference was the use of the reflection spectrum of skin without carrot juice as the reference white spectrum.

Measurements were attempted using both fluorescent and incandescent light bulbs inside, and using direct sunlight as light source outside. The object measured was the skin of my forearm with and without carrot juice.

3.2.2 Bandpass filters

An alternative to measuring all wavelengths is to measure only the wavelengths needed to calculate the bilirubin concentration. This can be achieved using optical bandpass filters. Two bandpass filters were bought from Thorlabs. The filters have a central transmit wavelength of 450(FB450-40, Thorlabs Inc, USA) and 550(FB550-40, Thorlabs Inc, USA) nanometers, and a full width half maximum of 40nm. These two central wavelengths were chosen because they are very close to two isosbestic points of oxygenated and deoxygenated hemoglobin, which are at 452 and 545nm[16]. In addition, bilirubin contributes to absorption around 450 nanometers, but the absorption is practically zero above 500nm(See Figure 3.1). This means that a measurement can be done for the light absorption of blood and skin using the bandpass filter with central wavelength at 550nm. Similarly, the light absorption of blood, skin and bilirubin can be measured using the other bandpass filter with a central wavelength at 450nm. The bilirubin concentration can then theoretically be calculated by subtracting the effect of blood and skin from the measurement at 450nm.

Measurements were done by holding the bandpass filter in front of the camera lens while capturing images. The white balance mode of the camera was set to something other than auto before the images were captured, similarly to how it was done for the foldable spectrometers. Images using both bandpass filters were captured for regular skin, for a skin bruise, and for skin with carrot juice applied. The green and blue color values of the images were then analyzed to see whether the bandpass filters could distinguish between regular skin and skin with increased bilirubin concentration or carrot juice. The green color value was used for the images captured with the 550nm filter, and the blue value was used with the 450nm filter.

3.2.3 Linear relationship between color and bilirubin

A researcher in Thailand has had some success measuring the bilirubin concentration of Thai newborn using a digital camera[49]. Because the paper is written in Thai, the method used is not entirely clear. But through the English summary and by looking at the images in the article, a highly probable method can be deciphered.

In the paper, newborn were photographed with a strip of paper next to them containing several squares colored in shades of gray ranging from black to white.

The images were then white-balance corrected in Photoshop using one of the gray squares on the colored paper strip. In addition, the black part of the color strip was set as RGB color value (0,0,0) in the images using the color level tool in Adobe Photoshop. This was also done for the whitest part of the color strip, with the color set to (255,255,255), and finally for one of the gray squares, which was set to (128,128,128). After this, it seems that the images were converted to the CMYK color space through the L*a*b* color space. The final value was then calculated as the value of yellow minus magenta, or Y-M. This procedure was performed on 61 newborns, and a linear relationship between the value of Y-M and the total serum bilirubin concentration was found.

As a reproduction of this method, two sheets of paper were printed. One black with the RGB color of (0,0,0), and one gray with the color (128,128,128). A standard paper sheet was used as the white (255,255,255) reference. Images were then captured of my arm with and without carrot juice, and with the three sheets of paper next to it. The images were then processed as described above using GIMP instead of Photoshop, with the only exception that the images were not transformed to the CMYK color space. The final result was instead calculated as G-B in the RGB color space. This corresponds to the Y-M calculation in the CMYK color space because Yellow, Y, corresponds to the mixture of red and green in RGB, while Magenta, M corresponds to the mixture of blue and red. The resulting calculation of Y-M in the CMYK color space therefore closely corresponds to the calculation of G-B in the RGB color space.

3.2.4 True color analysis

An alternative to taking pictures and blood samples of newborn and finding color correlations afterwards, is to attempt to predict the color of the newborns skin with varying levels of bilirubin. These predictions can then be compared to a measurements performed using a camera. This requires good numerical simulations to predict the skin colors, as well as cameras that are calibrated to reproduce the true colors of the captured scene. High-end digital cameras are today calibrated using images of e.g. a Macbeth ColorChecker in RAW format. Free tools are available that can perform such a calibration, but no smartphones on the market today support RAW file output. For this reason, a different calibration procedure was needed.

To calibrate the smartphone cameras, pictures of a Macbeth ColorChecker (MacBeth ColorChecker, Munsell Color, Baltimore, USA) were taken. Most of these images were captured using diffuse daylight through a window as light source. For these images, the colors given in a paper by Pascale [50] were used as the reference ColorChecker colors. Images were also captured using only the built-in flashlight of the smartphones as the light source. For those images, the reference ColorChecker colors were calculated from the reflection spectra of the ColorChecker and the light spectrum of the flashlights. The reflection spectra of the ColorChecker were gathered from the Munsell Color Science Laboratory website[51].

The pixel coordinates of the ColorChecker squares in the images were found manually. The color of the squares were then calculated as the average of all

the pixels in a square box centered on the squares pixel coordinate. The side length of these square boxes were set to 21 pixels, making the color of one of the ColorChecker squares the average of 441 pixels within that square. These colors were then converted to the xyY color system as outlined in 2.3, as they could then be compared with the reference ColorChecker values.

Before calibration was attempted, the standard deviation of the color reproduction of the cameras was measured. This was done by taking 10 pictures of the ColorChecker from slightly different angles using diffuse daylight through a window as light source. Series of ten pictures were taken using both daylight and auto white balance mode for both cameras. The colors of the squares were then found and converted to xyY as described above. An estimate of the standard deviation of the xy chromaticity was then found by averaging the standard deviation of the x and y values of the 24 individual ColorChecker squares. The final standard deviation was then calculated as the vector sum of the x and y standard deviations, $\Delta xy = \sqrt{\Delta x^2 + \Delta y^2}$.

Two calibration procedures were developed. The first is a generalized polynomial transform, which was reported as precise by researchers in the field of computer vision[34]. The other method is an implementation of the Thin-Plate Spline interpolation algorithm, which has been reported as a highly efficient calibration technique[35]. The theory behind the techniques is described in section 2.5. Both methods were developed to be used for three color channels, i.e. the RGB and XYZ color spaces. The Thin-Plate Spline interpolation algorithm was later modified to also work using only two color channels, so that it could be used for calibration of xy chromaticity. The general polynomial transform was not modified in the similar way because the Thin-Plate Spline method had proven to be superior.

The two methods efficiencies were tested using ordinary cross-validation. Ordinary cross-validation works by leaving one test sample out when creating a prediction model such as a general polynomial transform. The prediction model is then used to predict the value of the test sample that was left out during the models creation. The error of this prediction therefore becomes an estimate of the accuracy of the prediction model. This procedure is repeated leaving out a different test sample each repetition until all test samples have been left out once. The prediction errors of these repetitions are then averaged to give the estimate of the final models prediction error. In the case where the test samples are the 24 colors of the ColorChecker, 24 prediction models were created leaving one different color out for each model. These 24 models were then used to predict the value of the color that was left out during the creation of the model. The average of the errors of these predictions were then used as an estimate of the error of the final prediction model created using all 24 colors.

Ordinary cross-validation was also used to optimize the calibration methods. For the general polynomial transform, the polynomial order could be changed to test for example whether a higher order would yield a more precise calibration. Finding the optimal order was quick and easy because increasing the order above three more often than not decreased the quality of the calibration. Thus, finding the optimum polynomial order involved running ordinary cross-validation for order

one, two and three and finding the polynomial order with the smallest error. The Thin-Plate Spline method on the other hand, needed fine-tuning of the smoothing parameter λ which can be any number above or equal to zero(see equation 2.25). An iterative procedure was developed that first tested a range of values of λ using ordinary cross validation. After this, a new range of values were tested centered on the value λ that showed the least error in the previous iteration. Using this method, a highly accurate estimation of the optimal value of λ was found after only 6 or 7 such iterations. The matlab scripts used for the color calibration can be seen in appendix A.3, and the scripts for the iteration procedure can be found in appendix A.3.2.1.

The calibration methods were tested on images taken using diffuse daylight through a window as light source. Pictures of the ColorChecker were taken using all the different white balance settings on the cameras to find the best option. At the same time, pictures were taken of my bruised arm so that the calibration method could be compared to a real case of increased bilirubin as well as simulations. The colors of the pictures of my arm were then corrected using the calibrations calculated from the images of the ColorChecker. Then the pixel coordinates of two points on my arm were found manually. First, a point where there was a clear yellow color from my bruise, and second, a point on my arm with no bruise or other clearly visible pigments such as moles. The color of these two points were then calculated as the average color of the colors inside a square with side lengths of 21 pixels centered around the points. These colors were then converted to xy chromaticities in order to compare them to numerical simulations and the calibration quality.

Calibration was, as mentioned earlier, also performed on pictures of the ColorChecker captured using the built-in flashlight on the smartphones as the only light source. This was done because the reference colors of the ColorChecker given in the paper by Pascale[50] were calculated using D50 as light source. D50 is a good daylight simulator, but it is highly likely that there is some deviation from the true colors of the ColorChecker. To calculate the true colors of the ColorChecker using the flashlights, their light spectra were measured using the SD2000 spectrometer with the SpectraSuite computer application. A 10 millisecond integration time was used along with smoothing by using the average of ten spectra as the resulting spectrum. In addition, the color spectrum of the flashlight on an Iphone 5 was also measured to see if there is a large variance in different smartphone flashlights. These spectra could then be combined with the reflection spectra of the ColorChecker[51] using equations 2.15 and 2.16 to get the true colors of the ColorChecker.

3.2.5 Camera lens pressed against skin

There are apps available on the app markets today that are able to measure the pulse of a person by pressing the camera and built-in flashlight of the smartphones against the skin of a finger[52]. This could possibly be used to measure bilirubin concentrations as well. This technique has not been studied extensively during this project, but a couple of pictures were taken while pressing the lens and flashlight against the skin with a light pressure. Similarly to the foldable spectrometer

method, the white balance of the cameras were set to a non-auto setting before capturing the images.

Chapter 4

Results

4.1 Foldable spectrometer

Neither of the two spectrometers could be used with incandescent light bulbs. A spectrum was seen when the spectrometer was pointed directly at the bulbs, but when measurements were attempted on objects illuminated by the bulbs, the light sensitivity of the spectromteters proved inadequate. The same issue became apparent with the spectrometer with the smallest slit opening when it was used with fluorescent light bulbs. Only the major peaks in the fluorescent light spectrum created a response in the captured image. The spectrometer with a larger slit opening fared better using fluorescent light bulbs, but as can be seen in Figure 4.1, the reflection is practically zero where there are no major peaks. It was hoped that the peaks of the light source spectrum would disappear when the attenuation spectra of my arm were calculated. But this proved not to be the case, as can be seen in Figure 4.2. The peaks of the fluorescent light bulb are still seen dominating the spectrum.

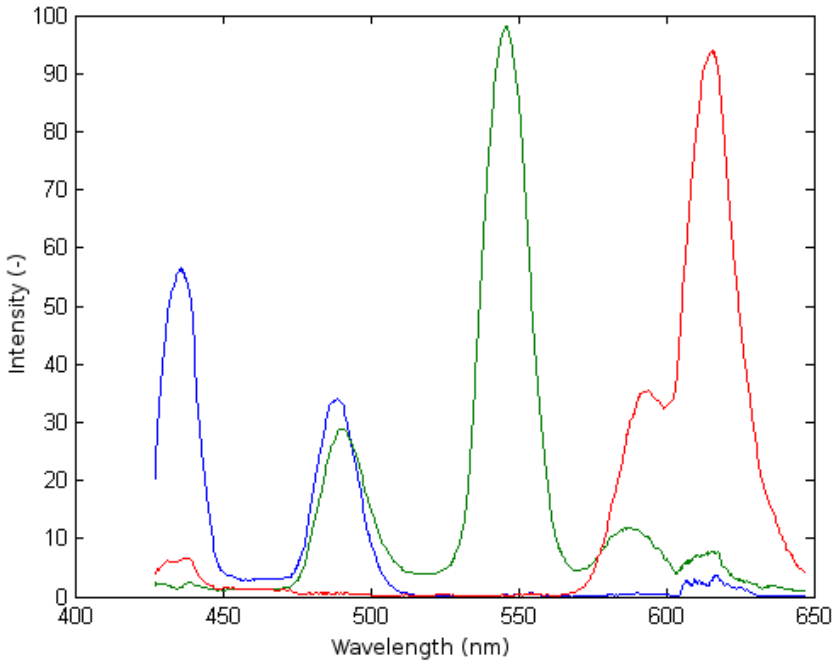


Figure 4.1: Reflection spectrum of a white sheet of paper using a fluorescent light bulb as light source. This spectrum was measured using the foldable spectrometer with the large slit opening. Keep in mind that the red, green, and blue lines represents the RGB channels of the image.

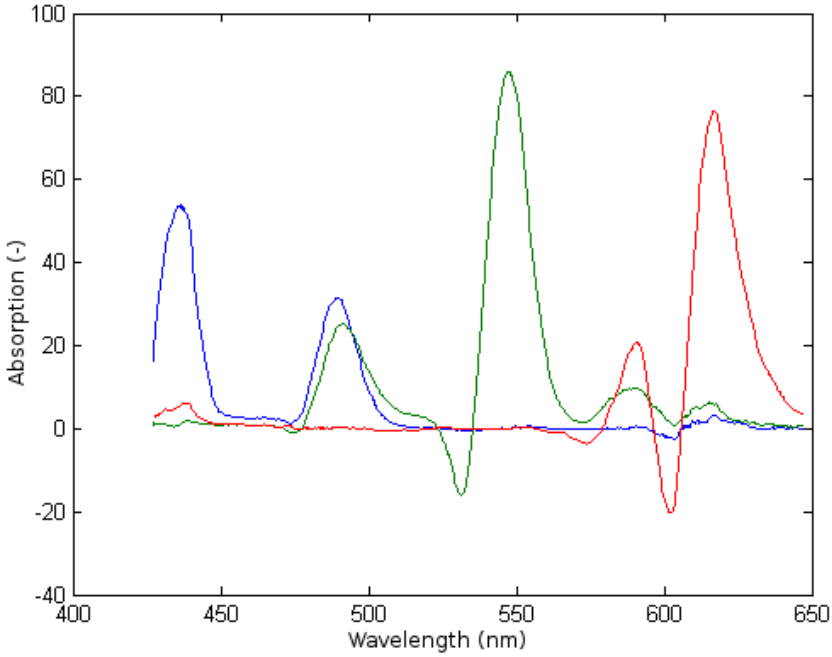


Figure 4.2: Attenuation spectrum of my arm with carrot juice using a fluorescent light bulb as light source. This spectrum was measured using the foldable spectrometer with the large slit opening.

When using direct sunlight as light source, the spectrometer with the small slit opening could be used to measure reflectance spectra. When possible, this spectrometer is preferred because it yields a higher resolution than the one with the larger slit opening. The measured attenuation spectrum of the carrot juice applied to my arm can be seen in Figure 4.3. This attenuation spectrum was measured and calculated in the same way as the attenuation spectrum of carrot juice in Figure 3.1. A peak between 450 and 510 nanometers can be seen in both spectra, but keep in mind that the wavelength calibration of the foldable spectrometer is not highly accurate. A negative value in these attenuation spectra indicate an increase in reflection when the carrot juice was applied. A sharp peak of this kind can be seen at 600nm for the measurement using the foldable spectrometer. This is not seen for the spectrum in Figure 3.1, where a slight increase in reflection is seen from 600 to 700 nanometers.

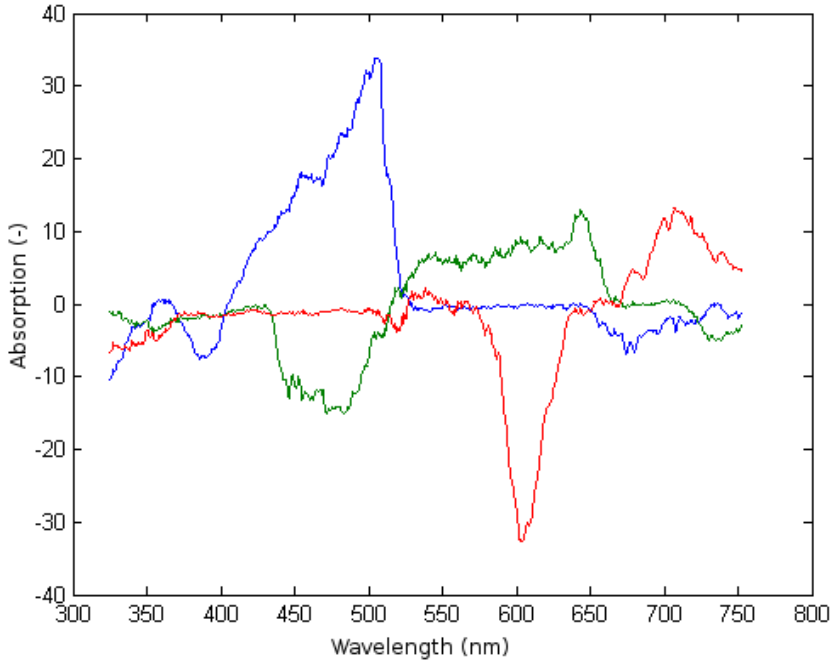


Figure 4.3: Attenuation spectrum of carrot juice on my arm using direct sunlight as light source. This spectrum was calculated in the same way as the attenuation spectrum of carrot juice seen in Figure 3.1, and was measured using the foldable spectrometer with the small slit opening.

4.2 Bandpass filters

The images captured using the HTC One V with the 550nm bandpass filters were all saturated (value 255) for the green color channel. The images captured with the 450nm filter were not saturated, but are of no use when they can't be compared to the images using the 550nm filter.

The Samsung fared better, although high values of the green channel were recorded using the 550nm filter for this camera as well. For the measurements of my skin without any bruise or carrot juice, the green value using the 550nm filter was 202, while the blue value using the 450nm filter was 201. For the measurements of my skin bruise, the green value using the 550nm filter was 208, while the blue value using the other filter was 192. For the measurements using carrot juice, the green value was measured to be 192, and the blue value was 158. Both the measurements of my skin with carrot juice and a bruise show a clear drop in value from the measurement using the 550nm filter to the one using the 450nm filter. This drop is not seen in the measurements of my regular skin.

4.3 Linear relationship between color and bilirubin

A clear increase in the value of green minus blue was found for skin applied with carrot juice compared to my regular skin. For the HTC, the value of green minus blue increased from 2 to 28, and for the Samsung the value increased from 22 to 33. This increase was backed by results from numerical simulations, which also showed an increase in the value of green minus blue when the amount of bilirubin was increased. The large difference between the results of the HTC and the Samsung is probably due to the different color responses of the cameras since the measurements were done using the same camera settings within a small time window of each other.

4.4 True color analysis

The standard deviation of the color reproduction of the HTC was estimated to be 0.0036 for the auto white balance setting in xy color coordinates. For the daylight white balance setting, it was estimated to 0.0034. Both white balance settings resulted in a standard deviation of 0.0016 using the Samsung. These standard deviations are illustrated in Figure 4.4, where the standard deviation is seen as the radius of the circles in the figure. The blue lines in the figure represent series of simulations with increasing concentrations of bilirubin, from 0 micromolar at the bottom left of the line to 200 micromolar at the top right. Each blue line to the right of another is another series of simulations of increasing bilirubin, but for a higher concentration of melanin. The absorption of melanin at 694nm is $250\,\mathrm{m}^{-1}$ for the leftmost line, and $2000\,\mathrm{m}^{-1}$ for the rightmost line.

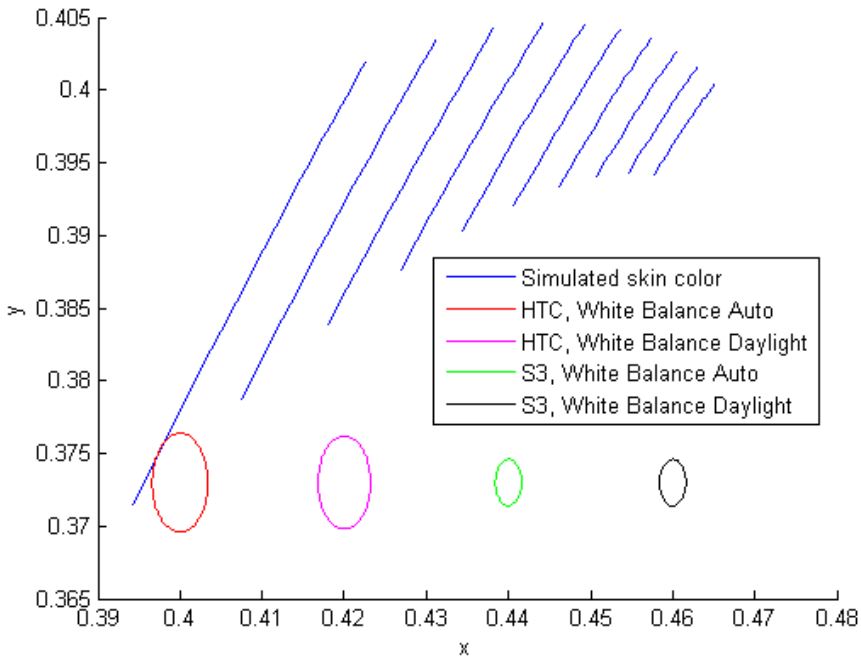


Figure 4.4: The standard deviation of the cameras plotted with skin simulation results. The radius of the circles are set to the respective cameras standard deviation. The blue lines are skin simulations for increasing levels of bilirubin. Each connected blue line represents a series of 20 simulations of increasing bilirubin for a single pigmentation level. Simulated pigmentation levels are higher for the blue lines farther to the right in the figure.

The size of the circles in Figure 4.4 can be compared to the length of the blue lines as an indicator of whether the cameras color variance is low enough to be able to separate skin with high concentrations of bilirubin from skin with lower concentrations. This appears to be true for both cameras except for the HTC when compared to simulations with very high melanin concentrations.

Before color calibration was performed on the images, the average error of the measured colors of the ColorChecker was usually between 50 and 60, measured as the vectorial color distance in RGB color space. Using the general polynomial transform to color calibrate the images, this error was reduced to approximately 30. The Thin-Plate Spline interpolation on the other hand, managed to decrease the error to approximately 25. Using the Thin-Plate Spline interpolation on the images in xy coordinates yields prediction errors of 0.0126 and 0.0275 for the HTC with white balance set to auto and daylight respectively. For the Samsung the prediction errors were 0.0157 and 0.0163 with the same white balance settings. These values are plotted in Figure 4.5 similarly to the standard deviations plotted in Figure

4.4. The reason for the large value of the calibration error of the HTC with white balance set to daylight is not known. Prediction errors were also calculated for the other white balance settings, but auto and daylight were the ones with the best results for the images captured with diffuse daylight through a window as light source.

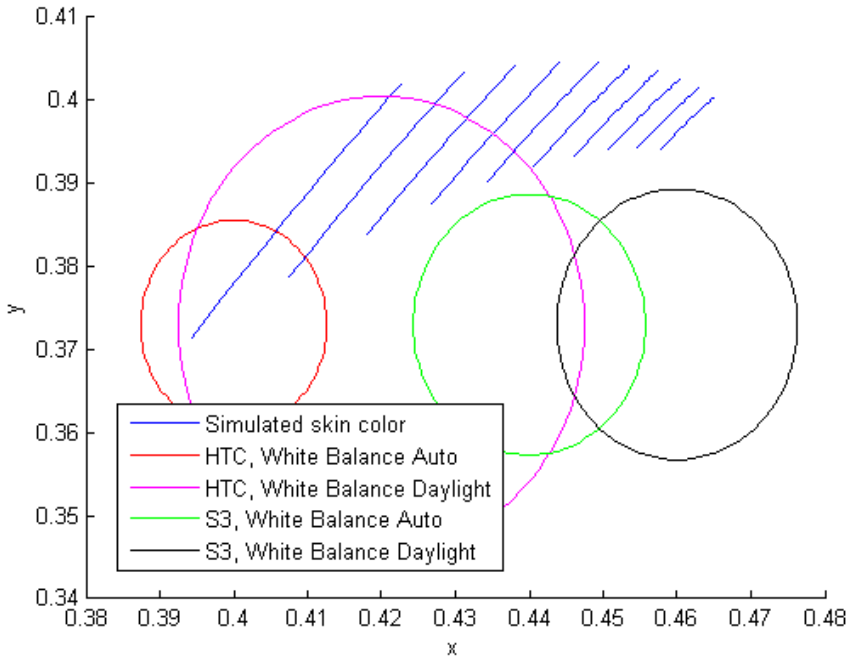


Figure 4.5: The predicted error of the calibration plotted with skin simulation results. The radius of the circles are set to the respective predicted calibration error. The blue lines are skin simulations for increasing levels of bilirubin, as in Figure 4.4.

The diameter of the circles in Figure 4.5 are of approximately the same length as the length of the blue lines. This indicates that this calibration procedure is not precise enough for bilirubin concentrations to be estimated from comparing a color measurement to numerical simulations. A calibration improvement was seen when the images taken using the flashlight of the phones as light source were used. When the reference colors from the paper by Pascale[50] were used to calibrate the images taken using the flashlights, the results were prediction errors of 0.015 and 0.023 for the HTC and Samsung respectively. Using the colors of the ColorChecker calculated by combining the flashlight spectra and the reflection spectra of the ColorChecker, calibration errors were reduced to 0.012 and 0.018. An improvement of approximately 20%.

The measured flashlight spectra can be seen in Figure 4.6. All spectra are very similar and share the same characteristic shape. This makes it likely that most smartphones use the same type of diodes in their flashlights, sharing a similar spectrum.

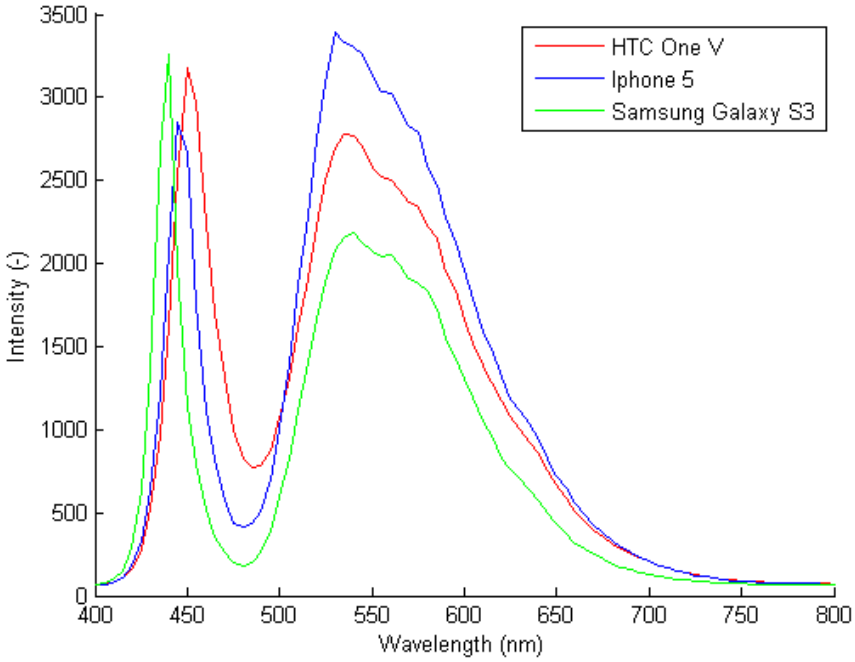


Figure 4.6: Light intensity spectra of the flashlights of three different phones.

Color measurements of my skin with and without a bruise were also performed using the same calibrations as were used for the prediction errors shown in Figure 4.5. These measurements can be seen in Figure 4.7. Each circle represents a single color measurement. Red circles indicate a measurement done on skin with no bruise. Green circles indicate measurements done on a bruise. The lines connecting them indicate what camera and white balance setting was used. The blue lines represent simulations as before. All the lines going from a measurement of my skin to a measurement of my skin with a bruise point up to the right similarly to the numerical simulations of increasing bilirubin concentrations.

The red circles are all measurements of the same skin. This is also true of the green circles, which are measurements of the same skin bruise. In an ideal case, all the red circles would be placed at the same coordinate and similar for all the green circles. The calibration errors illustrated in Figure 4.5 are the reasons for the spread of the measurements. In spite of this, two of the measurements can be seen very close to each other. But this is most likely a mere coincidence.

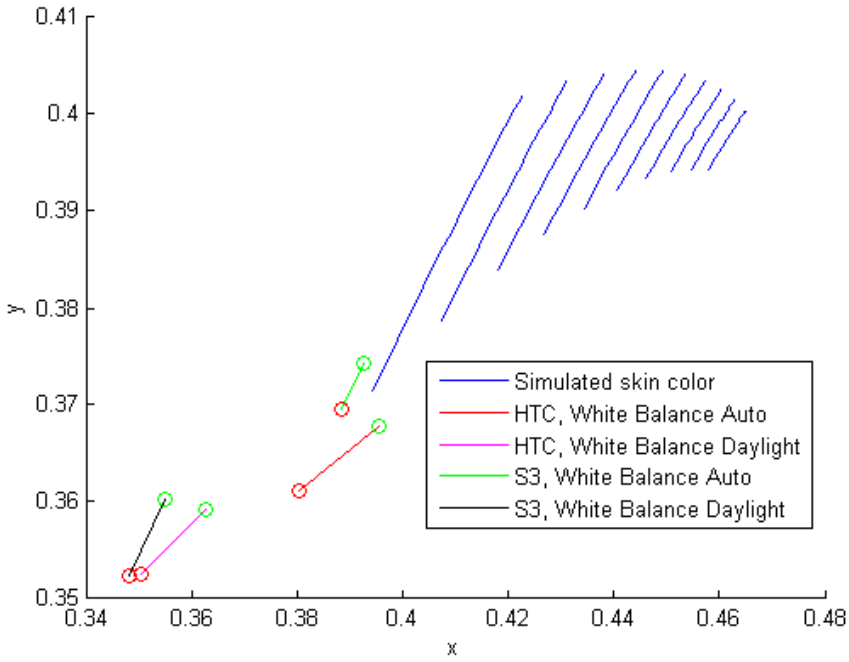


Figure 4.7: Color measurements of my skin plotted with skin simulation results. Red circles indicate measurement of my skin with no bruise. Green circles indicate measurements of my skin with yellow color from a bruise. Lines connecting them indicate the camera and white balance setting used in the measurements. The blue lines are skin simulations for increasing levels of bilirubin, as in Figure 4.4.

Numerical simulations, such as the ones seen in the above figures, were also performed with the blood volume fractions of the two top skin layers of the model set to 2%, instead of 1%. The results of these simulations can be seen in Figure 4.8 along with the results of simulations with 1% blood volume fraction. The series of simulations with higher blood volume fraction is the series of red lines. The top right value of the first red line is almost at the same coordinate as the bottom left of the fourth blue line. The xy coordinate of this point is approximately (0.427, 0.388). This indicates that the chromaticity alone cannot be used to give a single estimate of the bilirubin concentration. A measurement of the light intensity could solve this issue. The light intensity, or Y coordinate in the xyY color space, of the blue and red points at the coordinate mentioned above was 0.212 and 0.295 respectively. These light intensity values were normalized with the intensity of the light source set to 1.

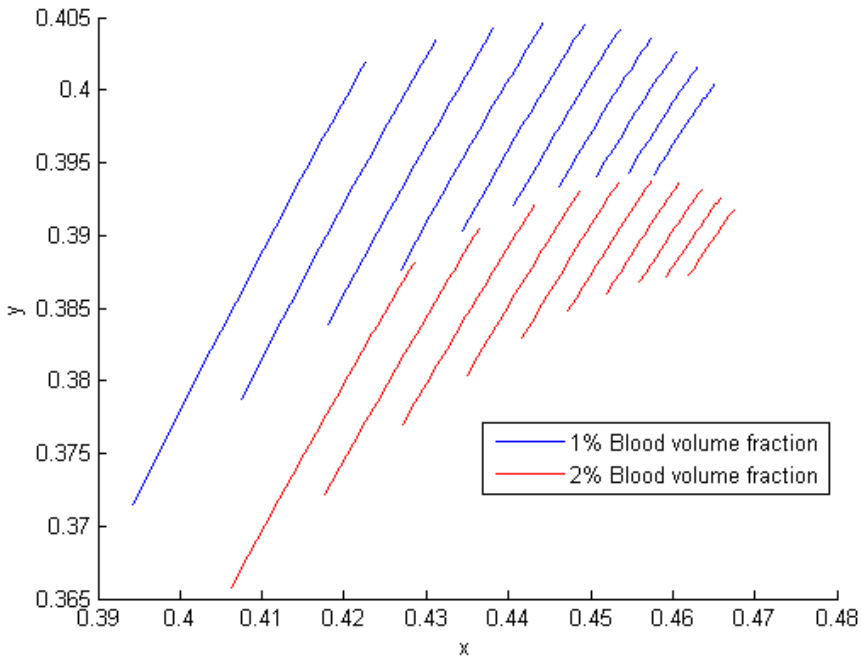


Figure 4.8: Skin simulation results with varying blood volume fraction. The blue lines are skin simulations for increasing levels of bilirubin with the blood volume fraction set to 1%, as in Figure 4.4. The red lines are the same except for an increased blood volume fraction to 2%.

4.5 Camera lens pressed against skin

The resulting image from pressing the camera lens and flashlight of the Samsung smartphone to my finger showed a near saturation of red over the whole image. In addition, a blue color gradient was seen with the highest blue value of 100 in the part which was closest to the flashlight. The blue value dropped to 37 in the part of the image farthest away from the flashlight. A similar gradient was seen for the green channel, but much smaller, with a drop from 15 to 0. The HTC produced an image with similar qualities, except that the value of the blue channel was zero over the whole image, and the green color gradient ranged from 153 nearest to the flashlight, to 2 on the opposite side of the image.

Chapter 5

Discussion

5.1 Foldable spectrometer

Obtaining a full reflectance spectrum of a newborn would be ideal when the measurement of bilirubin concentration is attempted. The reflectance spectrum can, as mentioned earlier, be used for more than bilirubin measurements[23], perhaps making diagnosis of other diseases or disorders possible. But the foldable spectrometers tested do not seem like a viable technology for such measurements. Even though the measurements of the light attenuation of carrot juice in Figure 4.3 showed decent overlap with the same measurements done with a proper spectrometer(Figure 3.1), the spectrometers have certain flaws that would limit their use in a practical setting.

First of all, the results show that the spectrometers need very bright light in order to function. This will in most settings limit the measurements to be done outside in the sun. This makes the procedure tedious and less likely to be performed. Second, the spectrometers need wavelength calibration each time they are attached to the phone, which also contributes to making the measurements tedious. In addition, the availability of fluorescent light bulbs, as was used for calibration in this project, is not guaranteed in a low resource setting. This calibration technique was also very inaccurate, perhaps too inaccurate to be used for actual bilirubin measurements.

Those are the main issues of the foldable spectrometers. The issue of reattaching the spectrometers could be solved by developing a more robust attachment procedure. The issue of wavelength calibration could perhaps be solved by using the spectrum of sunlight. The peak with the highest intensity in the solar spectrum is known as a function of the time of day. The problem would be to find other known wavelengths in the solar spectrum that can be used together with the main peak for calibration. The solar spectrum contains several peaks and valleys at known wavelengths, but the foldable spectrometers do not provide a spectrum with a high enough resolution to be able to detect those peaks and valleys. This

could be resolved by using a spectrometer with a smaller slit opening, but that would in turn require even more intense light to perform measurements.

The spectrometers would also need to be calibrated for intensity before they could be used for actual measurements. This could be done by shining light with a known spectral distribution onto the spectrometers. The resulting measured spectrum could then be adjusted according to the known spectrum of the light source. This procedure would have had to be done for at least each smartphone model that was to be used, since the spectral responses of the different camera models are expected to differ significantly.

5.2 Bandpass filters

The results of the measurements using the bandpass filters indicate that it is possible to measure bilirubin concentrations with this technique. However, further work would be needed to map the color values of the measurements to bilirubin concentrations. Different smartphone models would also need different color mappings. In addition, varying pigmentation levels in newborn makes it highly likely that a different color mapping would be needed for different skin types. The melanin has a higher absorption at shorter wavelengths, which could severely distort measurements at 450nm. Color saturation was also an issue with the HTC, but it is unknown how many smartphones would encounter similar problems.

In addition to these issues, the main issue with this method, in my opinion, is the price of the filters. Each filter costs 90 US dollars, and similar filters were found on eBay at approximately half the prize. A sum of 90 dollars is not too expensive for hospitals and health care institutions in low-income countries, but this limits the potential use of this method to such institutions. In addition, the price of the other methods tested during this project all fall below 10 dollars, making this method the most expensive by far. If cheap filters can be produced, and an easy method of attaching and detaching the filters to the camera is developed, then this method could potentially be successful.

5.3 Linear relationship between color and bilirubin

The results of the measurements with carrot juice indicate that the cameras are able to measure the difference in skin color due to increased concentration of bilirubin. This is further backed by the results in Figures 4.4 and 4.7. Figure 4.4 shows that the standard deviations of the cameras color responses are small enough that the changes in skin color due to bilirubin will not drown in a sea of noise when captured by the cameras. Figure 4.7 shows that all cameras and settings managed to measure the increased concentrations of bilirubin due to a skin bruise. These results, along with the positive results in the article from Thailand[49] makes this method promising. But there are a few problems with this method as well.

One of the problems is that it does not correct for varying pigmentation levels in the newborn. This method will therefore only work when performed on newborn with a similar skin type as the newborn studied when developing the method[22]. In addition, all cameras that are to be used for measurements will need to be calibrated by taking several pictures of newborn, as in the article from Thailand[49]. This is because different cameras have different color responses. The article and the results in this project show that cameras can indeed be used to measure bilirubin levels, but a more sophisticated approach is needed if the goal is a tool that can be used for all skin types and all smartphones.

5.4 True color analysis

The purpose of this method was to see whether the cameras could be used to measure the true colors of the scene it captured. These captured colors could then be compared to numerical simulations of skin color to give an estimate of the bilirubin concentration of the skin. Figure 4.4 shows that the standard deviation of the cameras is small enough that this should be possible. If this had not been the case, several images of the target could have been captured and the means of the captured color values could have been used instead. This technique can of course be used when the standard deviation is small as well, as a means of reducing the errors caused by the camera itself. The results of the calibration procedures in Figure 4.5 on the other hand, shows a far larger error than can be accepted if bilirubin concentrations are to be measured. A possible remedy for this is suggested by Menesatti et al.

Menesatti et al.[35, p. 12] reports that when using the thin-plate spline interpolation method, the use of ColorCheckers with more color patches results in calibrations of higher accuracy than calibrations performed with fewer color patches. In addition, they showed that the farther away the measured color was from the closest reference patch color, the larger the calibration error. For this reason, they suggest the use of ColorCheckers with 24-30 colors that closely match the colors of interest to reduce calibration error. ColorCheckers could even be created and printed *ad hoc*, and the colors could then be measured *a posteriori* using a spectrometer.

Numerical skin simulations could be used to calculate the whole range of skin colors expected to exist in newborn. A ColorChecker could then be created with many color patches within that color range. In this way, the calibration error could quite possibly be reduced to within an acceptable level in the color region of interest.

The color response of the cameras should also be tested using different light intensities of the same light source. If the cameras produce the same xy chromaticity values under varying light intensities, the cameras will only need to be calibrated once for each light source. Also, this would allow the chromaticity to be measured separately from the light intensity. The light intensity could be measured by holding something with a known reflectance, e.g. a white sheet of paper, next to the newborn as an intensity reference. This would simplify the measuring procedure because only the right type of light source would be needed, and not a certain

intensity of said light source. Knowing the reflected light intensity in addition to the chromaticity would help in the estimation of the bilirubin concentration. This was illustrated in Figure 4.8, where there were two almost equal chromaticity values of simulations with very different bilirubin concentrations. But the simulated light intensities at this point differed by approximately 40%, making it possible to separate these two chromaticity measurements with a measurement of the light intensity.

It would be very convenient if calibration only had to be performed on a single smartphone of a certain model, and all smartphones of that model could then use the same calibration. Knowing if this is possible is of great interest since it would simplify the distribution of the tool greatly. If this is not possible, every smartphone that is to be used as a diagnostic tool will need to be calibrated. This would make it much harder for the tool to reach distant low-resource settings. The variance of the color response of different smartphones of the same model can be estimated by for example taking pictures of ColorCheckers under different light conditions using many different smartphones of that specific model type. If the variance is then shown to be too large, each smartphone would have to be calibrated separately.

This method could possibly also be used as a diagnostic tool for other conditions. As mentioned in section 2.1.3, the reflection spectrum of newborn skin can be used to calculate more than the bilirubin concentration. The color of the skin is a result of the reflection spectrum and could therefore be used to calculate some of the same parameters, even though most of the spectral information is lost when the reflection spectrum is collapsed to three color values. Takiwaki and Serup[53] found such color indicators on psoriatic plaque, opening the possibility of using color analysis to diagnose psoriasis, or at least to measure the extent of it. But the technique is not limited to the medical field. Menesatti et al.[35] mentions color evaluation by biologists in field studies and quality control of food processing as possible applications of the method. Because of the wide variety of possible applications of this method, further studies should be conducted to find other appropriate practical cases where it can be implemented.

5.5 Camera lens pressed against skin

The results gathered for this method do not really convey more information other than that it might be possible to measure bilirubin concentrations using it. The concentration could for example be calculated from the gradient of the blue or green color of the image. For skin types with more pigmentation, the red value of the image might decrease because of the increased absorption from melanin in the skin. This could yield a way of distinguishing skin of varying skin pigmentation, removing the need of calibration for all various skin types. Numerical simulations of how far the light of different wavelengths will travel through skin with varying properties could also be developed. This could be used to estimate the amount of bilirubin needed in the skin to create a blue or green color gradient as is seen in the resulting images. Such ideas should be explored further, but that has not been done during this project.

Chapter 6

Conclusion

Of all the methods tested during this project, the method named true color analysis is the most promising. All the other methods, except pressing the camera lens to the skin, have certain flaws that are believed to either create inaccurate measurements or lower the probability that they will be used in a low-resource setting. The foldable spectrometers have issues with light intensity and calibration, the bandpass filters are expensive compared to the other methods, and the linear relationship between the measured yellow color and bilirubin concentrations will most likely only hold for a homogenous population.

The true color analysis method circumvents these issues by not requiring any external equipment, and by the possibility of correcting for varying pigmentation using reflected light intensity measurements. This method could therefore potentially work for all skin types and populations. Also, if tests of different smartphones of the same model, such as different Samsung Galaxy S3's, show little variation in color response, the calibration of all S3's could be performed by calibrating a single phone. This would greatly simplify the distribution of the tool, allowing it to be downloaded as an app, fully functional after download.

The method of pressing the camera lens onto the skin should also be mentioned as a worthy candidate method. Since this method won't require any external equipment either, implementation in low-resource settings would be simplified. The preliminary tests show that the method could potentially work, but it has not been tested thoroughly enough in this project for a confident recommendation to be given.

6.1 Recommendations for further work

It is recommended that the method of true color analysis is further developed. The main problem of this method in its current state is the color calibration error, which is far too large. This problem could be solved by creating a new ColorChecker with 30 color patches where most of the color patches have a chromaticity similar to

the chromaticity of both jaundiced and healthy newborn skin. These chromaticity values can be found by numerical simulations of skin with varying pigmentation, blood and bilirubin levels.

The color responses of the cameras should also be tested using different light intensities of the same light source. If the cameras reproduce the same chromaticity values with varying light intensities, calibration will only have to be performed once for each light source. Light intensity values could also be measured separately in the same image by holding something with known reflectance close to the object measured.

It is also of great interest to know whether every smartphone will need calibration, or if calibration of a single smartphone model will hold for all smartphones of that model. It is therefore recommended to acquire several different phones of the same model. The color response of the models could be tested by taking pictures of a ColorChecker under varying light conditions. The variation in color response between the cameras will then hopefully not be much larger than the standard deviation of the color response of a single camera, allowing calibration to be performed on a single phone of a certain model.

For this method to be a success when estimating bilirubin concentrations, numerical simulations of skin will have to be performed for all ranges of the parameters, such as bilirubin, that have an effect on the resulting skin color. This needs to be done in order to compare the color measured by the cameras to a result of the skin simulations. In this way, the bilirubin concentration can be found by looking at the concentration used when performing the simulation that resulted in the measured color. The simulations will need to be accurate if this method is expected to give good estimations of bilirubin concentrations. It should therefore be looked into whether the optical diffusion approximation yields accurate enough results, or if it should be replaced by Monte Carlo methods.

To finish off, the method of pressing the camera lens and flashlight onto the skin should be further developed if the method of true color analysis proves not to be adequate. In addition, research should be conducted to find other areas where the quantitative color analysis can be implemented. This should be done because there might be several other areas where high-cost analytical equipment can be at least partly replaced by cheap camera systems such as smartphones. This could greatly increase the availability of such tools in low income countries, potentially improving the quality of life of their people.

Bibliography

[1] Tina M. Slusher, Alvin Zipursky, and Vinod K. Bhutani. A global need for affordable neonatal jaundice technologies. *Seminars in Perinatology*, 35(3):185–191, June 2011. Available from: `http://linkinghub.elsevier.com/retrieve/pii/S0146000511000437`, `doi:10.1053/j.semperi.2011.02.014`.

[2] Vinod K. Bhutani, Alvin Zipursky, Hannah Blencowe, Rajesh Khanna, Michael Sgro, Finn Ebbesen, Jennifer Bell, Rintaro Mori, Tina M. Slusher, Nahed Fahmy, Vinod K. Paul, Lizhong Du, Angela A. Okolo, Maria-Fernanda de Almeida, Bolajoko O. Olusanya, Praveen Kumar, Simon Cousens, and Joy E. Lawn. Neonatal hyperbilirubinemia and rhesus disease of the newborn: incidence and impairment estimates for 2010 at regional and global levels. *Pediatric Research*, 74:86–100, December 2013. Available from: `http://www.nature.com/doifinder/10.1038/pr.2013.208`, `doi:10.1038/pr.2013.208`.

[3] Tina M Slusher, Bolajoko O Olusanya, Hendrik J Vreman, Ronald J Wong, Ann M Brearley, Yvonne E Vaucher, and David K Stevenson. Treatment of neonatal jaundice with filtered sunlight in nigerian neonates: study protocol of a non-inferiority, randomized controlled trial. *Trials*, 14:446, 2013. `doi:10.1186/1745-6215-14-446`.

[4] V A Moyer, C Ahn, and S Sneed. Accuracy of clinical judgment in neonatal jaundice. *Archives of pediatrics & adolescent medicine*, 154(4):391–394, April 2000.

[5] A Carceller-Blanchard, J Cousineau, and E E Delvin. Point of care testing: transcutaneous bilirubinometry in neonates. *Clinical biochemistry*, 42(3):143–149, February 2009. `doi:10.1016/j.clinbiochem.2008.09.106`.

[6] Karin Källander, James K Tibenderana, Onome J Akpogheneta, Daniel L Strachan, Zelee Hill, Augustinus H A ten Asbroek, Lesong Conteh, Betty R Kirkwood, and Sylvia R Meek. Mobile health (mHealth) approaches and

lessons for increased performance and retention of community health workers in low- and middle-income countries: a review. *Journal of medical Internet research*, 15(1):e17, 2013. doi:10.2196/jmir.2130.

[7] Lise Lyngsnes Randeberg. *Diagnostic applications of diffuse reflectance spectroscopy*. PhD thesis, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 2005. Available from: http://ntnu.diva-portal.org/smash/get/diva2:125853/FULLTEXT01.pdf.

[8] A N Bashkatov, E A Genina, V I Kochubey, and V V Tuchin. Optical properties of human skin, subcutaneous and mucous tissues in the wavelength range from 400 to 2000 nm. *Journal of Physics D: Applied Physics*, 38(15):2543–2555, August 2005. Available from: http://stacks.iop.org/0022-3727/38/i=15/a=004?key=crossref.33282f52c8be33f7c28513be248e8c27, doi:10.1088/0022-3727/38/15/004.

[9] Children, third degree burns and how to recognize 6 degrees of burns. Available from: http://pattyinglishms.hubpages.com/hub/Treating-Young-Children-Injured-by-Third-Degree-Burns.

[10] R. Rox Anderson and John A. Parrish. The optics of human skin. *Journal of Investigative Dermatology*, 77(1):13–19, July 1981. Available from: http://www.nature.com/jid/journal/v77/n1/abs/5615637a.html, doi:10.1111/1523-1747.ep12479191.

[11] Fitzpatrick TB. THe validity and practicality of sun-reactive skin types i through vi. *Archives of Dermatology*, 124(6):869–871, June 1988. Available from: http://dx.doi.org/10.1001/archderm.1988.01670060015008, doi:10.1001/archderm.1988.01670060015008.

[12] M. L. Wolbarsht, A. W. Walsh, and G. George. Melanin, a unique biological absorber. *Applied Optics*, 20(13):2184–2186, 1981. Available from: http://www.opticsinfobase.org/ao/fulltext.cfm?uri=ao-20-13-2184.

[13] L. T. Norvang, T. E. Milner, J. S. Nelson, M. W. Berns, and L. O. Svaasand. Skin pigmentation characterized by visible reflectance measurements. *Lasers in Medical Science*, 12(2):99–112, June 1997. Available from: http://link.springer.com/article/10.1007/BF02763978, doi:10.1007/BF02763978.

[14] Rashmi Sarkar, Srikanta Basu, R. K. Agrawal, and Piyush Gupta. Skin care for the newborn. *Indian pediatrics*, 47(7):593–598, 2010. Available from: http://link.springer.com/article/10.1007/s13312-010-0132-0.

[15] Rachel Ash-Bernal, Robert Wise, and Scott M. Wright. Acquired methemoglobinemia: A retrospective series of 138 cases at 2 teaching hospitals. *Medicine September 2004*, 83(5):265–273, 2004. Available from: http://ovidsp.ovid.com/ovidweb.cgi?T=JS&CSC=Y&NEWS=N&PAGE=fulltext&D=ovftg&AN=00005792-200409000-00001.

[16] Janelle E. Phelps, Karthik Vishwanath, Vivide T. C. Chang, and Nirmala Ramanujam. Rapid ratiometric determination of hemoglobin concentration using UV-VIS diffuse reflectance at isosbestic wavelengths. *Optics Express*, 18(18):18779–18792, August 2010. Available from: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3093134/`, `doi:10.1364/OE.18.018779`.

[17] A F McDonagh and D A Lightner. 'Like a shrivelled blood orange'–bilirubin, jaundice, and phototherapy. *Pediatrics*, 75(3):443–455, March 1985.

[18] N. E. I. Langlois and G. A. Gresham. The ageing of bruises: A review and study of the colour changes with time. *Forensic Science International*, 50(2):227–238, September 1991. Available from: `http://www.sciencedirect.com/science/article/pii/037907389190154B`, `doi:10.1016/0379-0738(91)90154-B`.

[19] Neville R. Pimstone, Raimo Tenhunen, Paul T. Seitz, Harvey S. Marver, and Rudi Schmid. The enzymatic degradation of hemoglobin to bile pigments by macrophages. *The Journal of Experimental Medicine*, 133(6):1264–1281, June 1971. Available from: `http://jem.rupress.org/content/133/6/1264`, `doi:10.1084/jem.133.6.1264`.

[20] T. W. R. Hansen and D. Bratlid. Bilirubin and brain toxicity. *Acta Pædiatrica*, 75(4):513–522, July 1986. Available from: `http://onlinelibrary.wiley.com/doi/10.1111/j.1651-2227.1986.tb10242.x/abstract`, `doi:10.1111/j.1651-2227.1986.tb10242.x`.

[21] Vinod K. Bhutani, Glenn R. Gourley, Saul Adler, Bill Kreamer, Chris Dalin, and Lois H. Johnson. Noninvasive measurement of total serum bilirubin in a multiracial predischarge newborn population to assess the risk of severe hyperbilirubinemia. *Pediatrics*, 106(2):e17–e17, August 2000. Available from: `http://pediatrics.aappublications.org/content/106/2/e17`.

[22] Steven L Jacques, David G Oelberg, and Iyad Saidi. Method and apparatus for optical measurement of bilirubin in tissue, October 1994. Available from: `http://www.google.com/patents/US5353790`.

[23] L. Lyngsnes Randeberg, E. Bruzell Roll, L. T. Norvang Nilsen, T. Christensen, and L. O. Svaasand. In vivo spectroscopy of jaundiced newborn skin reveals more than a bilirubin index. *Acta Pædiatrica*, 94(1):65–71, 2005. Available from: `http://onlinelibrary.wiley.com/doi/10.1111/j.1651-2227.2005.tb01790.x/abstract`, `doi:10.1111/j.1651-2227.2005.tb01790.x`.

[24] Robin M. Pope and Edward S. Fry. Absorption spectrum (380?700 nm) of pure water. II. integrating cavity measurements. *Applied Optics*, 36(33):8710–8723, November 1997. Available from: `http://ao.osa.org/abstract.cfm?URI=ao-36-33-8710`, `doi:10.1364/AO.36.008710`.

[25] L. O. Svaasand, L. T. Norvang, E. J. Fiskerstrand, E. K. S. Stopps, M. W. Berns, and J. S. Nelson. Tissue parameters determining the visual appearance

of normal skin and port-wine stains. *Lasers in Medical Science*, 10(1):55–65, March 1995. Available from: http://link.springer.com/article/10.1007/BF02133165, doi:10.1007/BF02133165.

[26] Akira Ishimaru. Diffusion of light in turbid material. *Applied Optics*, 28(12):2210, June 1989. Available from: http://www.opticsinfobase.org/ao/fulltext.cfm?uri=ao-28-12-2210&id=32482, doi:10.1364/AO.28.002210.

[27] Richard C. Haskell, Lars O. Svaasand, Tsong-Tseh Tsay, Ti-Chen Feng, Matthew S. McAdams, and Bruce J. Tromberg. Boundary conditions for the diffusion equation in radiative transfer. *JOSA A*, 11(10):2727–2741, 1994. Available from: http://www.opticsinfobase.org/abstract.cfm?id=841.

[28] Thorsten Spott and Lars O. Svaasand. Collimated light sources in the diffusion approximation. *Applied Optics*, 39(34):6453, December 2000. Available from: http://8.18.37.105/ao/abstract.cfm?uri=ao-39-34-6453, doi:10.1364/AO.39.006453.

[29] T. Smith and J. Guild. The C.I.E. colorimetric standards and their use. *Transactions of the Optical Society*, 33(3):73, January 1931. Available from: http://iopscience.iop.org/1475-4878/33/3/301, doi:10.1088/1475-4878/33/3/301.

[30] CIE downloads, 2014. Available from: http://www.cie.co.at/.

[31] ir-photo.net, 2014. Available from: http://www.ir-photo.net/ir_imaging.html.

[32] Creative commons licenses, 2014. Available from: https://creativecommons.org/licenses/.

[33] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3):95–99, 1976.

[34] Adrian Ilie and Greg Welch. Ensuring color consistency across multiple cameras. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, page 1268–1275. IEEE, 2005. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1544866.

[35] Paolo Menesatti, Claudio Angelini, Federico Pallottino, Francesca Antonucci, Jacopo Aguzzi, and Corrado Costa. RGB color calibration for quantitative image analysis: The "3D thin-plate spline" warping approach. *Sensors*, 12(12):7063–7079, May 2012. Available from: http://www.mdpi.com/1424-8220/12/6/7063/, doi:10.3390/s120607063.

[36] Fred L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989. doi:10.1109/34.24792.

[37] Malcolm H. Davis, Alireza Khotanzad, Duane P. Flamig, and Steven E. Harms. A physics-based coordinate transformation for 3-d image matching. *Medical Imaging, IEEE Transactions on*, 16(3):317–328, 1997. Available from: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=585766`.

[38] Marcello Pelillo and Edwin R. Hancock. *Energy Minimization Methods in Computer Vision and Pattern Recognition: International Workshop EMMCVPR'97, Venice, Italy, May 21-23, 1997, Proceedings*. Springer, April 1997.

[39] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Prof Dr Walter Schempp and Prof Dr Karl Zeller, editors, *Constructive Theory of Functions of Several Variables*, number 571 in Lecture Notes in Mathematics, pages 85–100. Springer Berlin Heidelberg, January 1977. Available from: `http://link.springer.com/chapter/10.1007/BFb0086566`.

[40] Grace Wahba. *Spline Models for Observational Data*. SIAM, September 1990.

[41] TEST: samsung galaxy s III (s3) - multimedia. Available from: `http://www.amobil.no/artikler/samsung-galaxy-s-iii-s3/109432`.

[42] TEST: HTC one v - multimedia. Available from: `http://www.amobil.no/artikler/htc-one-v/110020`.

[43] Molar extinction coefficient of bilirubin in chloroform. Available from: `http://omlc.ogi.edu/spectra/PhotochemCAD/data/119-abs.txt`.

[44] Noboru Ohta and Alan Robertson. *Colorimetry: fundamentals and applications*. John Wiley & Sons, 2006. Available from: `http://books.google.com/books?hl=en&lr=&id=U8jeh1uhSHgC&oi=fnd&pg=PR2&dq=%22standard+color+matching+functions+in+1931+based%22+%22when+expressed+in+radiometric%22+%22functions,+the+CIE+adopted+an+average+of+the+data%22+%22COLOR+SPECIFICATION%22+%22first+sight,+this+seems+strange+because+a+negative+value%22+&ots=SU2nCQkQmV&sig=kGZ9phQl8Z2ZoI3d28sgT2D0hD0`.

[45] Foldable mini-spectrometer. Available from: `http://store.publiclab.org/products/foldable-mini-spectrometer`.

[46] Foldable spectrometer phone. Available from: `http://publiclab.org/wiki/foldable-spec`.

[47] Convert spectrographs to spectrograms. Available from: `http://jethomson.wordpress.com/spectrometer-articles/convert-spectrographs-to-spectrograms/`.

[48] Compact FLuorescent light bulb spectrum, 2014. Available from: `http://commons.wikimedia.org/wiki/File:Fluorescent_lighting_spectrum_peaks_labelled.png`.

[49] Somsak Leartveravat. Transcutaneous bilirubin measurement in full term neonate by digital camera. *Medical Journal of Srisaket Surin Buriram Hospitals*, 24(1):105 – 118, 2009. Available from: `http://thailand.digitaljournals.org/index.php/MJSSBH/article/view/5427`.

[50] Danny Pascale. RGB coordinates of the macbeth ColorChecker. 2006. Available from: `http://www.babelcolor.com/download/RGB%20Coordinates%20of%20the%20Macbeth%20ColorChecker.pdf`.

[51] Useful color data. Available from: `http://www.rit.edu/cos/colorscience/rc_useful_data.php`.

[52] Heart beat counter android. Available from: `https://play.google.com/store/apps/details?id=mobisy.heartbeat.counter`.

[53] Hirotsugu Takiwaki and Jørgen Serup. Measurement of color parameters of psoriatic plaques by narrow-band reflectance spectrophotometry and tristimulus colorimetry. *Skin Pharmacology and Physiology*, 7(3):145–150, 1994. Available from: `http://www.karger.com/Article/Abstract/211289`, `doi:10.1159/000211289`.

Appendices

Appendix A

Matlab scripts

A.1 Wavelength calibration

This is the wavelength calibration script used for the calibration of the foldable spectrometers. It was modified to work with the foldable spectrometers by changing the variables REF_PEAK_WAVELENGTHS and PEAKNUM so that the script would work for only two reference peaks. The original script can be found at Jonathan Thomsons website[47]. His code was released using the BSD Open source license, allowing the usage and modification of his work.

```
1  % This script has been edited by Gunnar Vartdal
2
3  % 0001 %WAVELENGTH_CALIBRATE - Takes a horizontal slice of a ...
       reference spectrograph,
4  % 0002 %                       and through user interaction finds ...
       the pixel locations
5  % 0003 %                       that correspond to known wavelengths ...
       which are then used
6  % 0004 %                       to calculate and output a calibrated ...
       wavelength scale.
7  % 0005 %
8  % 0006 % It's not necessary to use every row of a spectrograph to ...
       derive the
9  % 0007 % spectrogram. Although every column is always used. To ...
       specify a horizontal
10 % 0008 % slice (region), use the optional arguments y0 and h.
11 % 0009 %
12 % 0010 % This function uses the mercury peaks found in a compact ...
       fluorescent lamp's
13 % 0011 % (CFL) spectrum as a wavelength reference. It is possible to...
        use other
14 % 0012 % reference peaks (e.g. neon) with this function by slightly ...
       modifying it.
15 % 0013 %
16 % 0014 % Syntax:  lambda = wavelength_calibrate(img, roi)
```

```
17 % 0015 %
18 % 0016 % Inputs:
19 % 0017 %    img - an image matrix or an image's filename.
20 % 0018 %    roi - region of interest of spectrograph to process.
21 % 0019 %
22 % 0020 % Outputs:
23 % 0021 %    lambda - a calibrated wavelength scale in nanometers.
24 % 0022 %
25 % 0023 % Example:
26 % 0024 %    lambda = wavelength_calibrate('path/filename');
27 % 0025 %    h=100; y0=(size(I,1)-h)/2; lambda = wavelength_calibrate...
       (I, [y0, h]);
28 % 0026 %
29 % 0027 % Other m-files required: image2spectrum
30 % 0028 % Subfunctions: none
31 % 0029 % MAT-files required: none
32 % 0030 % Other files required: CFL_Hg_peaks_labeled.png
33 % 0031 %
34 % 0032 % See also: IMAGE2SPECTRUM, POLYFIT
35 % 0033 %
36 % 0034 % Author: Jonathan Thomson
37 % 0035 % Work:
38 % 0036 % email:
39 % 0037 % Website: http://jethomson.wordpress.com
40 % 0038 %
41
42 function lambda = wavelength_calibrate(img, roi)
43
44     if (nargin < 1 || nargin > 2)
45         usage('wavelength_calibrate(img, roi)');
46     end
47
48 % 0046     % Look at CFL_Hg_peaks_labeled.png to see the numbered ...
       peaks and their
49 % 0047     % wavelengths. These are mercury (Hg) spectrum peaks. The...
        zeros in
50 % 0048     % REF_PEAK_WAVELENGTHS are for non-mercury peaks. Note ...
       that your CFL's
51 % 0049     % spectrogram may show a second weaker peak near 407.783 ...
       nm next to peak 1
52 % 0050     % at 404.656 nm. This is a mercury peak, but is not shown...
        in the image
53 % 0051     % CFL_Hg_peaks_labeled.png nor is it used for ...
       calibration.
54 % 0052
55 % 0053     %NIST Atomic Spectra Database
56 % 0054     %http://physics.nist.gov/PhysRefData/ASD/lines_form.html
57 % 0055     %REF_PEAK_WAVELENGTHS = [404.6565, 435.8335, 0, 0, 546...
       .0750, 0, 579.0670];
58 % 0056
59 % 0057     %NIST Handbook of Basic Atomic Spectroscopic Data
60 % 0058     %http://physics.nist.gov/PhysRefData/Handbook/Tables/...
       mercurytable2.htm
61 % 0059     %"This handbook is designed to provide a selection of the...
        most important
62 % 0060     %and frequently used atomic spectroscopic data in an ...
       easily accessible
```

```
63 % 0061     %format."
64     REF_PEAK_WAVELENGTHS = [435.8328, 0, 546.0735];
65 % 0063
66 % 0064     % This is a user defined constant, but peak numbers 2, 5,... and 7 should
67 % 0065     % work fine if you are using a CFL as the reference.
68     PEAKNUM = [1 3];
69
70     scrsz = get(0, 'ScreenSize');
71
72     if (¬exist('roi','var') || isempty(roi))
73     roi = [0, 0];
74     end
75
76     Z = image2spectrum(img, 'rgb', roi);
77
78 refpknm = REF_PEAK_WAVELENGTHS(PEAKNUM);
79
80     %if rng is large and peak 7 is too close to peak 6, peak 6 might... be
81     %incorrectly picked.
82     %rng = -10:10;
83     rng = -5:5;
84
85     dataisgood = false;
86     while (dataisgood == false)
87         n = figure('Position', [1 1 0.9*scrsz(3) 0.9*scrsz(4)]);
88         hold on
89         plot(fliplr(Z))
90         m = max(max(Z));
91         a = axis;
92         axis([a(1) a(2) 0 1.10*m])
93
94         for li = 1:length(PEAKNUM)
95             title(['Click on peak ' num2str(PEAKNUM(li)) '.']);
96             [x, ignored] = ginput(1); % 2nd output of ginput() is ... mandatory
97             x = round(x);
98
99             % The user cannot be expected to click exactly at the ... peak
100             % location so we should search on either side of gpxl
101             % (within a hardcoded range) to find the exact peak ... location.
102             dx = x + rng;
103
104             [peak_rgb, pkloc] = max(Z(dx,:));
105             [peak, chan] = max(peak_rgb); % which channel has the ... highest peak
106
107 % x-coordinate/column number of the peak
108             pkx(li) = dx(1) + pkloc(chan) - 1;
109
110             hold on, plot(pkx(li), peak, 'k*')
111         end
112
113         title(['The reference peaks'' locations have been recorded. ...
```

```
                ' ...
114                 'Please answer the question in the console.'])
115         rsp = input(['Did you pick the correct peaks? ' ...
116                      'If not, answer no to try again. [(y)es/(n)o/(q...
                             )uit]: '], 's');
117         if (¬isempty(rsp) && lower(rsp(1)) == 'y')
118             dataisgood = true;
119         elseif (¬isempty(rsp) && lower(rsp(1)) == 'q')
120             close(n)
121             error('wavelength_calibrate: instructed to quit by user....
                    ');
122         end
123
124         close(n)
125     end
126
127     if (dataisgood == true)
128         [p, s] = polyfit(pkx, refpknm, 1);
129         lambda = polyval(p, (1:size(Z,1)).');
130     end
131
132 end
```

A.2 Numerical skin simulation

Two scripts were used for the numerical simulation of skin, both written by Asgeir Bjørgan at the department of electronics and telecommunications at NTNU. He has given permission for them to be reproduced here.

The first is a script calculating the absorption and scattering coefficients of the different layers. This script has been modified by me, removing a bug and changing the input unit of the bilirubin concentration:

```
1  function [mua1, mus1, mua2, mus2, mua3, mus3, g] = calcSkinData(...
       lambda, oxy1, Bd1, oxy2, Bd2, muam694, wat1, wat2, wat3, fat1, ...
       fat3, bet1, bet3, met2, met3, bil2, bil3, aMie, aRay, bMie)
2      load('muab.mat');
3      muh_oxy_calc = @(x) interp1(muabo(:,1), muabo(:,2), x);
4      muh_deoxy_calc = @(x) interp1(muabd(:,1), muabd(:,2), x);
5
6
7      muh_oxy = muh_oxy_calc(lambda);
8      muh_deoxy = muh_deoxy_calc(lambda);
9      melanin_base = ((694./lambda)).^3.46;
10     g = 0.62 + lambda.*29e-5;
11
12     %absorption properties
13     H = 0.41;
14     H0 = 0.45;
15     Be = 0.002;
16     mua_other = 25;
17     muab_blood1 = (muh_oxy.*oxy1 + muh_deoxy.*(1-oxy1)).*H/H0;
18     muab_blood2 = (muh_oxy.*oxy2 + muh_deoxy.*(1-oxy2)).*H/H0;
```

```
19      mua_melanin = muam694.*melanin_base;
20
21      %absorption coefficients
22      mua1 = mua_melanin + muab_blood1.*Be + mua_other.*(1-Be);
23      mua2 = muab_blood1.*Bd1 + mua_other.*(1-Bd1);
24      mua3 = muab_blood2.*Bd2 + mua_other.*(1-Bd2);
25
26      %extra absorption from water and the like
27      muaw = load('hale73.dat');
28      muaf = load('fat.dat');
29      muabil = load('bil.dat');
30      muabet = load('bet.dat');
31      for i=1:length(lambda)
32          mua1(i) = mua1(i) + wat1.*water(lambda(i), muaw) + fat1.*fat...
                (lambda(i), muaf) + bet1.*bet(lambda(i), muabet);
33          mua2(i) = mua2(i) + wat2.*water(lambda(i), muaw) + bil2.*bil...
                (lambda(i), muabil);% + met2.*met(lambda(i));
34          mua3(i) = mua3(i) + wat3.*water(lambda(i), muaw) + fat3.*fat...
                (lambda(i), muaf) + bil3.*bil(lambda(i), muabil) + ...
                bet3.*bet(lambda(i), muabet);% + met3.*met(lambda(i));
35      end
36
37      %scattering properties
38
39      %bashkatov2005
40      must = 100.*(aMie.*(lambda./500.0).^-bMie + aRay.*(lambda./500)....
            ^-4);
41
42      mus1 = must./(1-g);
43      mus2 = must./(1-g);
44      mus3 = must./(1-g);
45  end
46
47  function w = water(x, muaw)
48      if ((x > 200) & (x < 90000))
49          w = interp1(muaw(:,1), muaw(:,2)*100, x, 'spline');
50      else
51          w = zeros(size(x));
52      end
53
54  end
55
56  function f = fat(x, muaf)
57      if ((x > 429) & (x < 1098))
58          f = interp1(muaf(:,1), muaf(:,2), x);
59      else
60          f = zeros(size(x));
61      end
62  end
63
64  % Endret av Gunnar til aa returnere m^-1/M fra cm^-1/M
65  function b = bil(x, muabil)
66      if ((x > 239) & (x < 510))
67          b = interp1(muabil(:,1), muabil(:,2), x);
68          b = 100*b;
69      else
70          b = zeros(size(x));
```

```
71      end
72  end
73
74  function b = bet(x, muabet)
75      if ((x > 219) & (x < 510))
76          b = interp1(muabet(:,1), muabet(:,2), x);
77      else
78          b = zeros(size(x));
79      end
80  end
```

The second script is the script calculating the diffuse reflection coefficient:

```
1   %isotropic source function, three-layer model
2   function r = ReflIsoL3(mua1, mus1, mua2, mus2, mua3, mus3, g, d1, d2...
        )
3       A = 0.17;
4       de = 100e-6;
5
6
7
8       %reduced scattering coefficients
9       musr1 = mus1.*(1.0-g);
10      musr2 = mus2.*(1.0-g);
11      musr3 = mus3.*(1.0-g);
12
13      %diffusion constant
14      D1 = 1.0./(3.0.*(musr1 + mua1));
15      D2 = 1.0./(3.0.*(musr2 + mua2));
16      D3 = 1.0./(3.0.*(musr3 + mua3));
17
18      %transport coefficients
19      mutr1 = musr1 + mua1;
20      u1 = mutr1;
21      mutr2 = musr2 + mua2;
22      u2 = mutr2;
23      mutr3 = musr3 + mua3;
24      u3 = mutr3;
25
26      %optical penetration depth
27      del1 = sqrt(D1./(mua1));
28      a1 = del1;
29      del2 = sqrt(D2./(mua2));
30      a2 = del2;
31      del3 = sqrt(D3./(mua3));
32      a3 = del3;
33
34      K1 = del1.*del1.*musr1./(D1.*(1-mutr1.*mutr1.*del1.*del1));
35      K2 = del2.*del2.*musr2./(D2.*(1-mutr2.*mutr2.*del2.*del2));
36      K3 = del3.*del3.*musr3./(D3.*(1-mutr3.*mutr3.*del3.*del3));
37      A2 = -(-2.*D2.*exp(-u1.*d1).*exp(u2.*(-d2+d1)).*(K3-K2).*D1.*D3....
            *a2-D1.*...
38      exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*a2.*D3.*K2+D1.*exp(-(u1.*d1.*...
            a2-d1+d2)./a2).*...
39      D2.*a2.*D3.*K1+D1.*exp(-(-d2+d1+u1.*d1.*a2)./a2).*D2.*a2.*D3.*K1...
            -D1.*...
```

```
40      exp(-(-d2+d1+u1.*d1.*a2)./a2).*D2.*a2.*D3.*K2-exp((-d2+d1)./a2)....
            *D3.*...
41      a2.*a2.*D1.*exp(-u1.*d1).*(D2.*u2.*K2-D1.*u1.*K1)+exp(-(-d2+d1)....
            /a2).*...
42      D3.*a2.*a2.*D1.*exp(-u1.*d1).*(D2.*u2.*K2-D1.*u1.*K1)+D1.*exp(-(...
            u1.*d1.*a2-d1+d2)./a2).*...
43      D2.*D2.*a3.*K2-D1.*exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*D2.*a3.*K1-...
            D1.*...
44      exp(-(-d2+d1+u1.*d1.*a2)./a2).*D2.*D2.*a3.*K2+D1.*exp(-(-d2+d1+...
            u1.*d1.*a2)./a2).*...
45      D2.*D2.*a3.*K1-2.*A.*a1.*D2.*exp(-u1.*d1).*exp(u2.*(-d2+d1)).*(...
            K3-K2).*...
46      D3.*a2-2.*exp(-u1.*d1-u2.*d2+u2.*d1).*a2.*D2.*D2.*u2.*K2.*a3.*D1...
            +A.*a1.*...
47      exp(-(-d2+d1+u1.*d1.*a2)./a2).*D2.*a2.*D3.*K1-A.*a1.*exp(-(-d2+...
            d1+u1.*d1.*a2)./a2).*...
48      D2.*a2.*D3.*K2+A.*a1.*exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*a2.*D3.*...
            K1-A.*a1.*...
49      exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*a2.*D3.*K2+A.*a1.*exp(-(-d2+d1...
            )./a2).*a3.*D2.*a2.*exp(-u1.*d1).*(D2.*u2.*K2-D1.*u1.*K1)+...
            A.*a1.*exp((-d2+d1)./a2).*D2.*a3.*a2.*...
50      exp(-u1.*d1).*(D2.*u2.*K2-D1.*u1.*K1)+2.*exp(-u1.*d1-u2.*d2+u2.*...
            d1).*a2.*...
51      D3.*K3.*u3.*a3.*D2.*D1+A.*a1.*exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*...
            D2.*a3.*...
52      K2-A.*a1.*exp(-(u1.*d1.*a2-d1+d2)./a2).*D2.*D2.*a3.*K1+A.*a1.*...
            ...
53      exp(-(-d2+d1+u1.*d1.*a2)./a2).*D2.*D2.*a3.*K1-A.*a1.*exp(-(-d2+...
            d1+u1.*d1.*a2)./a2)....
54      *D2.*D2.*a3.*K2+exp((-d2+d1)./a2).*D2.*a3.*a2.*D1.*exp(-u1.*d1)....
            *(D2.*u2.*K2-D1.*...
55      u1.*K1)+exp(-(-d2+d1)./a2).*a3.*D2.*a2.*D1.*exp(-u1.*d1).*(D2.*...
            u2.*K2-D1.*u1.*K1)-A.*...
56      a1.*exp((-d2+d1)./a2).*D3.*a2.*a2.*exp(-u1.*d1).*(D2.*u2.*K2-D1....
            *u1.*K1)+A.*...
57      a1.*exp(-(-d2+d1)./a2).*D3.*a2.*a2.*exp(-u1.*d1).*(D2.*u2.*K2-...
            D1.*u1.*K1)+2.*...
58      A.*a1.*exp(-u1.*d1-u2.*d2+u2.*d1).*a2.*D3.*K3.*u3.*a3.*D2-2.*A.*...
            a1.*...
59      exp(-u1.*d1-u2.*d2+u2.*d1).*a2.*D2.*D2.*u2.*K2.*a3-A.*a1.*a2.*...
            D2.*...
60      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*K1.*D3-A.*a1.*exp(-(d1.*a2...
            -d2.*a1+d1.*a1)./...
61      a1./a2).*a2.*D2.*K1.*D3+D1.*D1.*a2.*D2.*exp((-d1.*a2+d1.*a1-d2.*...
            a1)./a1./a2).*...
62      K1.*u1.*a3+D1.*a1.*D2.*D2.*exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2)....
            *K1.*u1.*a3+A.*...
63      D1.*a2.*D2.*exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*K1.*a3+D1.*D1....
            *...
64      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*a2.*D2.*K1.*u1.*a3+A.*D1.*...
            ...
65      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*a2.*D2.*K1.*a3-D1.*a1.*...
66      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D2.*D2.*K1.*u1.*a3+A.*D1.*...
            ...
67      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*a2.*a2.*K1.*D3+D1.*D1.*...
68      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*a2.*a2.*K1.*u1.*D3-D1.*D1....
            *...
```

```
69      a2.*a2.*exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*K1.*u1.*D3-A.*D1.*...
            a2.*a2.*...
70      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*K1.*D3+A.*a1.*D2.*D2.*...
71      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*K1.*a3-A.*a1.*exp(-(d1.*a2...
            -d2.*a1+d1.*a1)./...
72      a1./a2).*D2.*D2.*K1.*a3-D1.*a1.*a2.*D2.*exp((-d1.*a2+d1.*a1-d2.*...
            a1)./a1./a2).*...
73      K1.*u1.*D3-D1.*a1.*exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*a2.*D2....
            *K1.*u1.*D3).*...
74      a1./(exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*a1.*D2.*D2.*a3+...
            ...
75      exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*D1.*a2.*D2.*a3+...
76      exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*a1.*D2.*D2.*a3+...
77      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*a1.*D2.*D2.*a3-...
78      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*D1.*a2.*D2.*a3-...
79      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*a1.*D2.*D2.*a3+...
80      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*a1.*D2.*D2.*a3-...
81      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*D1.*a2.*D2.*a3-...
82      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*a1.*D2.*D2.*a3-...
83      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*a1.*D2.*D2.*a3+...
84      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*D1.*a2.*D2.*a3-...
85      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*a1.*D2.*D2.*a3+...
86      exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*a1.*D2.*a2.*D3+...
87      exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*a1.*D2.*a2.*D3+...
88      exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*D1.*a2.*a2.*D3+...
89      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*a1.*D2.*a2.*D3+...
90      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*a1.*D2.*a2.*D3-...
91      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*a1.*D2.*a2.*D3+...
92      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*D1.*a2.*a2.*D3-...
93      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*a1.*D2.*a2.*D3-...
94      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*D1.*a2.*a2.*D3+...
95      exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*a1.*D2.*a2.*D3-...
96      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*D1.*a2.*a2.*D3+...
97      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*a1.*D2.*a2.*D3+...
98      exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*D1.*a2.*D2.*a3+...
99      exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*D1.*a2.*D2.*a3+...
100     exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*A.*a1.*D1.*a2.*D2.*a3+...
101     exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*A.*a1.*D1.*a2.*D2.*a3+...
102     exp(-(-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*D1.*a2.*a2.*D3-...
103     exp((d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*D1.*a2.*a2.*D3-...
104     exp(-(d1.*a2-d2.*a1+d1.*a1)./a1./a2).*D1.*D1.*a2.*a2.*D3+...
105     exp((-d1.*a2+d1.*a1-d2.*a1)./a1./a2).*D1.*D1.*a2.*a2.*D3);
106     A1 = (A2.*(D1./del1-A)-K1.*(A+D1.*mutr1))./(D1./del1+A);
107     j = K1.*D1.*mutr1 + D1./del1.*A1 - A2.*D1./del1;
108     r = -j;
109 end
```

This is the script, written by me, used to calculate colors from the skin simulations:

```
1  %calculate_skin_color Calculates simulated color values of given ...
       skin parameters with given
2  %light source and observer
3  function [R, G, B, x, y, Y, diffusion_spectrum, reflected_spectrum] ...
       = calculate_skin_color(lambda, light_source, observer, oxy1, ...
       Bd1, oxy2, Bd2, muam694, wat1, wat2, wat3, fat1, fat3, bet1, ...
```

```
        bet3, met2, met3, bil2, bil3, aMie, aRay, bMie, d1, d2)
4
5
6       %Beregn hud-data
7       [mua1, mus1, mua2, mus2, mua3, mus3, g] = calcSkinData(lambda, ...
            oxy1, Bd1, oxy2, Bd2, muam694, wat1, wat2, wat3, fat1, fat3...
            , bet1, bet3, met2, met3, bil2, bil3, aMie, aRay, bMie);
8       %Beregn det reflekterte spekteret dividert paa innkommende ...
            intensitet
9       diffusion_spectrum = ReflIsoL3(mua1, mus1, mua2, mus2, mua3, ...
            mus3, g, d1, d2);
10      %Beregn det reflekterte spekteret med dagslys som lyskilde
11      reflected_spectrum = apply_light_source(lambda, ...
            diffusion_spectrum, light_source);
12
13      %Beregn fargen til lyskilden. Skal brukes til normalisering
14      light_source_spectrum = apply_light_source(lambda, ones(1, ...
            length(lambda)), light_source);
15      [Xs, Ys, Zs] = apply_observer(lambda, light_source_spectrum, ...
            observer);
16
17      %Beregn fargen til reflektert spekter og normaliser med ...
            lyskilden.
18      [X, Y, Z] = apply_observer(lambda, reflected_spectrum, observer)...
            ;
19      X = X/Ys;
20      Y = Y/Ys;
21      Z = Z/Ys;
22      x = X/(X+Y+Z);
23      y = Y/(X+Y+Z);
24
25      [R, G, B] = convert_XYZ_sRGB(X, Y, Z);
26
27  end
```

And these are the two helper functions used by the above script. Apply_light_source multiplies a given reflection spectrum with a light source spectrum. Apply observer calculates tristimulus color values using the given observer.

```
1   %Combines a reflection spectrum and a light source into a single ...
        spectrum
2   function Z = apply_light_source(lambda, spectrum, light_source)
3
4       Z = zeros(1, length(lambda));
5       for i=1:length(lambda)
6           b = interp1(light_source(:,1), light_source(:,2), lambda(i))...
                ;
7           Z(i) = b*spectrum(i);
8       end
9
10  end
```

```
1   %Applies an observer, e.g. CIE 2deg observer, to an intensity ...
        spectrum, resulting in
```

```
%tristimulus color data.
function [X, Y, Z] = apply_observer(lambda, spectrum, obs)

    X = 0;
    Y = 0;
    Z = 0;
    dl = lambda(2) - lambda(1);
    for i=1:length(lambda)
        xbar = interp1(obs(:,1), obs(:,2), lambda(i));
        ybar = interp1(obs(:,1), obs(:,3), lambda(i));
        zbar = interp1(obs(:,1), obs(:,4), lambda(i));
        X = X + spectrum(i)*xbar*dl;
        Y = Y + spectrum(i)*ybar*dl;
        Z = Z + spectrum(i)*zbar*dl;
    end

end
```

A.3 Color calibration

A.3.1 General polynomial transform

The general polynomial transform was calculated and applied using these two scripts:

```


%Creates transformation matrix with polynomial coefficients of the ...
    given order
%to transform source colors to the target color space
function Z = general_polynomial_transform(source_colors, ...
    target_colors, order)

    res_size = order*3 + 1;
    B = zeros(length(source_colors(:,1)), res_size);

    for i=1:length(source_colors(:,1))
        for j=1:3
            for k=1:order
                B(i,(j-1)*order + k) = source_colors(i,j)^k;
            end
        end
        B(i, res_size) = 1;
    end

    Z = pinv(B)*target_colors;

end
```

```


```

```
3  %Apply the general polynomial transform to the given color matrix
4  function Z = apply_GPT(GPT, colors)
5
6      order = (length(GPT(:,1))-1)/3;
7      B = zeros(length(colors(:,1)), length(GPT(:,1)));
8
9      for i=1:length(colors(:,1))
10         for j=1:3
11             for k=1:order
12                 B(i,(j-1)*order + k) = colors(i,j)^k;
13             end
14         end
15         B(i, length(GPT(:,1))) = 1;
16     end
17
18     Z = B*GPT;
19
20 end
```

Ordinary cross validation of the general polynomial transform method was performed with this script which returns the polynomial order giving the lowest prediction error along with the prediction error:

```
1  %Runs ordinary cross validation on the general polynomial transform ...
       method. Returns
2  %the order resulting in the calibration with the least error, along ...
       with the error.
3  function [order, color_difference] = ocv_gpt(source_colors, ...
       target_colors, start_order, stop_order)
4
5      n=length(source_colors(:,1));
6      test_source_colors = zeros(n-1, 3, n);
7      test_target_colors = zeros(n-1, 3, n);
8      test_source_colors(:,:,1) = source_colors(2:n,:);
9      test_target_colors(:,:,1) = target_colors(2:n,:);
10     for i=2:n-1
11         test_source_colors(1:(i-1),:,i) = source_colors(1:(i-1),:);
12         test_target_colors(1:(i-1),:,i) = target_colors(1:(i-1),:);
13         test_source_colors(i:n-1,:,i) = source_colors(i+1:n,:);
14         test_target_colors(i:n-1,:,i) = target_colors(i+1:n,:);
15     end
16     test_source_colors(:,:,n) = source_colors(1:n-1,:);
17     test_target_colors(:,:,n) = target_colors(1:n-1,:);
18
19     cnt = 1;
20     order(cnt) = start_order;
21     while(order(cnt) ≤ stop_order)
22         color_diff(cnt) = 0;
23
24         for i=1:n
25             GPT = general_polynomial_transform(test_source_colors...
                       (:,:,i), test_target_colors(:,:,i), order(cnt));
26             gpt_color = apply_GPT(GPT, source_colors(i,:));
27             color_diff(cnt) = color_diff(cnt) + get_color_diff(...
                       gpt_color, target_colors(i,:));
28         end
```

```
29
30          cnt = cnt +1;
31          order(cnt) = order(cnt-1)+1;
32      end
33
34      [C, I] = min(color_diff);
35      color_difference = C/(n-1);
36      order = order(I);
37
38  end
```

A.3.2 Thin-Plate Spline

The Thin-plate spline interpolation method was implemented and applied in this way for three dimensions:

```
1  %Calibrates the given colors in source_colors using target_colors as...
       the target color values
2  %and the thin-plate spline interpolation method. lambda is the ...
       smoothing parameter
3  function TPS = thin_plate_spline(source_colors, target_colors, ...
       lambda)
4
5      n = length(source_colors(:,1));
6
7      M = zeros(n,n);
8      T = zeros(n,4);
9      O1 = zeros(4,4);
10     O2 = zeros(4,3);
11
12     for i=1:n
13         T(i,1) = 1;
14         T(i,2) = source_colors(i,1);
15         T(i,3) = source_colors(i,2);
16         T(i,4) = source_colors(i,3);
17         for j=1:n
18            M(i,j) = Em(source_colors(i,:), source_colors(j,:));
19         end
20         M(i,i) = n*lambda;
21     end
22
23     A = [M, T; T' O1];
24     B = [target_colors; O2];
25     TPS = A\B;
26 end
```

```
1  %Applies thin plate spline interpolation to color data
2  function Z = apply_TPS(TPS, colors, source_colors)
3
4      M = 4;
5      n = length(TPS(:,1)) - M;
6
```

```
    Z = zeros(length(colors(:,1)), length(colors(1,:)));

    for i=1:length(colors(:,1))
        for c=1:3
            %TPS(n+1,c);
            Z(i,c) = Z(i,c) + TPS(n+1,c);
            for sc=1:3
                %TPS(n+1+sc,c)
                Z(i,c) = Z(i,c) + colors(i,sc)*TPS(n+1+sc,c);
            end
            for j=1:n
                %TPS(j,c)
                E = Em(colors(i,:), source_colors(j,:));
                Z(i,c) = Z(i,c) + TPS(j,c)*E;
            end
        end
    end

end
```

These are the scripts implementing and applying the thin-plate spline interpolation method in two dimensions:

```
%Same as thin_plate_spline except for two-dimensional chromaticity ...
    data
function TPS = thin_plate_spline_2d(source_chr, target_chr, lambda)

    n = length(source_chr(:,1));

    M = zeros(n,n);
    T = zeros(n,3);
    O1 = zeros(3,3);
    O2 = zeros(3,2);

    for i=1:n
        T(i,1) = 1;
        T(i,2) = source_chr(i,1);
        T(i,3) = source_chr(i,2);
        for j=1:n
           M(i,j) = Em_2d(source_chr(i,:), source_chr(j,:));
        end
        M(i,i) = n*lambda;
    end

    A = [M, T; T' O1];
    B = [target_chr; O2];
    TPS = A\B;
end
```

```
%apply_TPS Applies thin plate spline interpolation to chromaticity ...
    data
function Z = apply_TPS_2d(TPS, chromaticities, source_chr)

    M = 3;
```

```
    n = length(TPS(:,1)) - M;

    Z = zeros(length(chromaticities(:,1)), length(chromaticities...
        (1,:)));

    for i=1:length(chromaticities(:,1))
        for c=1:2
            %TPS(n+1,c);
            Z(i,c) = Z(i,c) + TPS(n+1,c);
            for sc=1:2
                %TPS(n+1+sc,c)
                Z(i,c) = Z(i,c) + chromaticities(i,sc)*TPS(n+1+sc,c)...
                    ;
            end
            for j=1:n
                %TPS(j,c)
                E = Em_2d(chromaticities(i,:), source_chr(j,:));
                Z(i,c) = Z(i,c) + TPS(j,c)*E;
            end
        end
    end

end
```

The functions Em and Em_2d seen in the above scripts corresponds to $U(r)$ in equation 2.27, and can be seen in these scripts:

```
%Em Calculates the Em function of TPS
function E = Em(t1, t2)

    tau = sqrt((t1(1)-t2(1))^2 + (t1(2)-t2(2))^2 + (t1(3)-t2(3))^2);
    %theta = -0.039789;
    %E = theta*tau;
    E = 2*(tau^2)*log(tau+1e-20);

end
```

```
%Em Calculates the Em function of TPS
function E = Em_2d(t1, t2)

    tau = sqrt((t1(1)-t2(1))^2 + (t1(2)-t2(2))^2);
    %theta = -0.039789;
    %E = theta*tau;
    E = 2*(tau^2)*log(tau+1e-20);

end
```

A.3.2.1 Ordinary cross validation for the thin-plate spline method

Ordinary cross validation of the thin-plate spline method was performed for three dimensions using this script:

```
%Runs ordinary cross validation on a series of values of lambda of ...
    the thin-plate
%spline interpolation method. Returns the value of lambda minimizing...
     the
%prediction error, and the prediction error itself
function [l, color_difference] = ocv_tps(source_colors, ...
    target_colors, start_lambda, stop_lambda, mult)

    n=length(source_colors(:,1));
    test_source_colors = zeros(n-1, 3, n);
    test_target_colors = zeros(n-1, 3, n);
    test_source_colors(:,:,1) = source_colors(2:n,:);
    test_target_colors(:,:,1) = target_colors(2:n,:);
    for i=2:n-1
        test_source_colors(1:(i-1),:,i) = source_colors(1:(i-1),:);
        test_target_colors(1:(i-1),:,i) = target_colors(1:(i-1),:);
        test_source_colors(i:n-1,:,i) = source_colors(i+1:n,:);
        test_target_colors(i:n-1,:,i) = target_colors(i+1:n,:);
    end
    test_source_colors(:,:,n) = source_colors(1:n-1,:);
    test_target_colors(:,:,n) = target_colors(1:n-1,:);

    cnt = 1;
    lambda(cnt) = start_lambda;
    while(lambda ≤ stop_lambda)
        color_diff(cnt) = 0;

        for i=1:n
            TPS = thin_plate_spline(test_source_colors(:,:,i), ...
                test_target_colors(:,:,i), lambda(cnt));
            tps_color = apply_TPS(TPS, source_colors(i,:), ...
                test_source_colors(:,:,i));
            color_diff(cnt) = color_diff(cnt) + get_color_diff(...
                tps_color, target_colors(i,:));
        end

        cnt = cnt +1;
        lambda(cnt) = lambda(cnt-1)*mult;
    end

    [C, I] = min(color_diff);
    color_difference = C/(n-1);
    l = lambda(I);

end
```

For two dimensions it was performed using this script:

```
function [l, chr_difference] = ocv_tps_2d(source_chr, target_chr, ...
    start_lambda, stop_lambda, mult)

    n=length(source_chr(:,1));
    test_source_chr = zeros(n-1, 2, n);
    test_target_chr = zeros(n-1, 2, n);
    test_source_chr(:,:,1) = source_chr(2:n,:);
    test_target_chr(:,:,1) = target_chr(2:n,:);
```

```
    for i=2:n-1
        test_source_chr(1:(i-1),:,i) = source_chr(1:(i-1),:);
        test_target_chr(1:(i-1),:,i) = target_chr(1:(i-1),:);
        test_source_chr(i:n-1,:,i) = source_chr(i+1:n,:);
        test_target_chr(i:n-1,:,i) = target_chr(i+1:n,:);
    end
    test_source_chr(:,:,n) = source_chr(1:n-1,:);
    test_target_chr(:,:,n) = target_chr(1:n-1,:);

    cnt = 1;
    lambda(cnt) = start_lambda;
    while(lambda ≤ stop_lambda)
        chr_diff(cnt) = 0;

        for i=1:n
            TPS = thin_plate_spline_2d(test_source_chr(:,:,i), ...
                test_target_chr(:,:,i), lambda(cnt));
            tps_chr = apply_TPS_2d(TPS, source_chr(i,:), ...
                test_source_chr(:,:,i));
            chr_diff(cnt) = chr_diff(cnt) + get_chromaticity_diff(...
                tps_chr, target_chr(i,:));
        end

        cnt = cnt +1;
        lambda(cnt) = lambda(cnt-1)*mult;
    end

    [C, I] = min(chr_diff);
    chr_difference = C/(n-1);
    l = lambda(I);

end
```

The iterative procedure returning the smoothing parameter, λ, minimizing the prediction error was implemented in these scripts for three and two dimensions:

```
%Iterates through different values of lambda, finding the value ...
    minimizing the
%prediction error of the thin-plate spline calibration
function [l, color_difference] = ocv_tps_iterative(source_colors, ...
    target_colors, start_lambda, stop_lambda, mult, iterations)

    [l, color_difference] = ocv_tps(source_colors, target_colors, ...
        start_lambda, stop_lambda, mult);

    for i=1:iterations-1
        start_lambda = l/mult;
        stop_lambda = l*mult;
        mult = 1 + mult/(10*i);
        [l, color_difference] = ocv_tps(source_colors, target_colors...
            , start_lambda, stop_lambda, mult);
    end

end
```

```
%Iterates through different values of lambda, finding the value ...
    minimizing the
%prediction error of the thin-plate spline calibration
function [l, chr_difference] = ocv_tps_2d_iterative(source_chr, ...
    target_chr, start_lambda, stop_lambda, mult, iterations)

    [l, chr_difference] = ocv_tps_2d(source_chr, target_chr, ...
        start_lambda, stop_lambda, mult);

    for i=1:iterations-1
        start_lambda = l/mult;
        stop_lambda = l*mult;
        mult = 1 + mult/(10*i);
        [l, chr_difference] = ocv_tps_2d(source_chr, target_chr, ...
            start_lambda, stop_lambda, mult);
    end

end
```