NTNU – Trondheim
Norwegian University of
Science and Technology

Estimation of Vital Signs from Ambient-Light Non-Contact Photoplethysmography

Nikolai Grov Roald

Master of Science in Electronics
Submission date: February 2013
Supervisor: Ilangko Balasingham, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Abstract

English

In this thesis we have investigated different aspects of non-contact photoplethysmography (PPG) using only ambient lighting. We have investigated how to develop a functional, automatic system based on this to detect heart rate. We have also investigated how to use the concept of non-contact PPG to acquire further relevant medical information from a human subject.

We have investigated different color spaces and found that the Hue and Saturation channels from HSL and HSV color spaces are far superior to the Green channel of the RGB color space, which has previously been used. Especially under circumstances with much noise, are these channels superior and more robust against noise.

The concept of independent component analysis (ICA) has been investigated as a method of improving results. It is found to improve some channels and color spaces, but the best ICA channel does not have better performance than the best non-ICA channel.

The phase of, and difference between, PPG signals has been investigated as a means of acquiring medical information. The phase measurements are highly vulnerable to noise, but there are indications that occlusion can induce a phase difference between different limbs. This difference can be used to calculate change in blood pressure.

We have synchronized ECG and PPG data, and found that there is a high correlation between the two. Pulse transit time (PTT) from the heart to the measurement site can be calculated using this synchronized information.

Further have different motion compensation algorithms and signal processing techniques been investigated with the goal of improving the PPG signal and a programs ability to automatically detect heart rate.

Norsk

I denne masteroppgaven har vi undersøkt forskjellige aspekter relatert til kontaktfri, fotoplehysmografi (PPG) ved bare bruk av omgivelseslys. Vi har undersøkt hvordan utvikle et funksjonelt, automatisk system basert på dette. Vi har også videre undersøkt hvordan man videre kan bruke konseptet kontaktfri PPG til å tilegne seg medisinsk informasjon om pasienter.

Vi har undersøkt forskjellige fargerom og funnet at Hue og Saturation fra fargerommene HSL og HSV er overlegen Grønn kanal fra RGB som tidligere har blitt brukt. Spesielt i situasjoner med mye støy er disse kanalene overlegne og mer robust mot støy.

Konseptet uavhengighetsanalyse (ICA) har blitt undersøkt som en metode for å forbedre resultat. Den kan forbedre resultat i noen kanaler, men den beste ICA kanalen gir fortsatt dårligere resultat enn den beste ikke-ICA kanalen.

Fasen til, og forskjellen mellom, PPG signal har blitt undersøkt som en metode for å tilegne seg medisinsk informasjon. Målingene har vist seg å være svært sårbare for støy, men tyder på at okkusjon kan lage en

faseforskjell mellom forskjellige lemmer. Denne forskjellen kan brukes til å beregne forandring i blodtrykk.

Vi har synkronisert EKG og PPG data, og har funnet en stor grad av korrelasjon i mellom dem. Pulsens gangtid fra hjertet til et målepunkt kan bli beregnet ved å bruke denne synkroniserte informasjonen.

Videre har forskjellige algorimter for bevegelses kompensasjon og signalbehandlings teknikker blitt undersøkt for i finne måter å forbedre PPG signalet og et programs mulighet til å automatisk detektere hjerterytme.

Problem Statement

It has been shown that photoplethysmography images can be used to estimate pulse and heart beat using advanced signal processing techniques. In this Master thesis, the candidate will perform a literature study to understand different signal processing techniques to improve the estimation of vital signs (pulse, heart rate, blood pressure, etc.). The emphasis will be given to understand the pulse transit time, phase variations, pressure variation as a function of time and differential pressure estimation.

The candidate will study methods to improve data acquisition from video, where emphasis will be given to reduce artifacts due to mechanical movements of the objects. Furthermore, different methods for compensation for light fluctuation during recording will be studied. The candidate will use a set of video images taken from persons with different skin colors, gender, age and under different lighting conditions.

The candidate will make assessment of different estimation techniques as well as clinically relevant information from the videos. In addition to differential blood pressure and pulse transit time between two extremities (hands, feet) and between extremities and the core of the body, the recordings will be correlated with an EKG measurement to extrapolate clinically relevant information.

The thesis should contain literature study, materials and methods, experimental results, discussions and conclusions in addition to references and computer source codes.

Contents

List of Figures

List of Abbreviations

HR Heart Rate

BP Blood Pressure

RR Respiration Rate

PPG Photoplethysmography

ABP Arterial Blood Pressure

RGB Red Green Blue color space

HSL Hue Saturation Lightness color space

HSV Hue Saturation Value color space

ICA Independent component analysis

JADE Joint Approximate Diagonalization of Eigenmatrices

SURF Speeded Up Robust Features

PTT Pulse transit time

ROI Region Of Interest

"I'll be honest, we're throwing science at the wall here,
to see what sticks. No idea what it'll do."
-Cave Johnson

1 Introduction

1.1 Motivation

The single largest cause of premature death in the developed world today is heart and circulatory disease. Unfortunately can these conditions go unnoticed for a long time before they suddenly manifest themselves, with potential lethal results. Had these conditions been detected earlier, lives would be saved and many tragedies could have been avoided.

In newspapers one can often read the sentence "had it been detected earlier" following bad news. Unfortunately today, detecting the early signs of heart diseases, requires invasive and expensive observation performed over a long period of time, usually by a medical professional. And due to the conditional nature of such a disease, and the erratic nature of the symptoms, there is no guarantee of detection. Therefore, such observations are usually not performed before other, more noticeable and sometimes terminal symptoms have manifested themselves. To perform such observation on all members of the public, with its resource intensive nature, is still not a viable option.

Therefore, the development of a cheap, non-invasive and easily available technology for use by the general public, that could detect the early signs of heart disease, could have a large positive effect on the common health.

1.2 Background

A common saying in science is that standing on the shoulders of giant makes you see longer and this is highly applicable to this thesis. The overarching idea, which the technology in this thesis is based on, is the theory that one can use video to detect human vital signs. As with the best ideas, it is simple but with large ramifications.

This theory has been investigated and improved on by others, and the work in this thesis is an attempt to further develop and to explore the subject. This thesis is in large part based on the works by Poh[23] and the master thesis of Rustand[28]. We will take the ideas and techniques from these works and attempt to further improve on them.

The problem statement in this thesis can be boiled down to two questions. The first is "how can one most efficiently measure human vital signs with video using only ambient lighting?" and the second is "what furher can we use this technology for?". In the course of this thesis we will attempt to answer both.

1.3 Outline of the works

The topic in this thesis can be divided into two themes in an attempt to answer the questions stated earlier. To answer the first we have developed a software framework for testing different techniques and ideas. We have explored the idea that other color spaces might be more efficient than the normal RGB color space at detection vital signs. We have also investigated and attempted to answer

other questions relating to the efficiency. “What is the optimal recording length for heart rate detection?”, “How does the video quality affect the results?”, “Can one create an algorithm for tracking and compensating for movement” and “How do recording settings affect the results, and what are the optimal circumstances”.

To answer the second theme question, of what further this technology can be used for, we have investigated the concept of phase difference between pulse waves, and to what extent we can use this information to obtain information about the body. We have synchronized the waves with ECG measurement to investigate what information can be obtained regarding synchronization and potential time delay between signals. Lastly we have investigated if video can be used to detect dangerous cells such as cancer or to detect poor blood circulation following organ transplants.

I would like to thank my supervisor Ilangko Balashingham of NTNU and Rikshospitalet for his advice and guidance and the people working at the intervention center at Rikshospitalet for their help during experiments performed there.

1.4 Report Structure

This report starts with a theory chapter on page 5 which describes the most important background information needed to understand the rest of the report. The theory chapter is divided into several parts. An explanation of the human vital signs we are trying to measure, a general description of photoplethysmography and the physics behind the signal are given in 2.1. Explanation of the color spaces used and the transformation between them are given in 2.3. Sections 2.4 and 2.5 are used to describe different algorithms used in this thesis.

In the preliminary chapter starting on page 19 we have a preliminary exploration of the subjects surrounding photoplethysmography. In 3.1 we will evaluate some of the previous works done on this subject. In 3.2 we will explain the video quality measurement and classifications we have used in the results section while in 3.3 we describe some of the video parameters that are important for recordings made for photoplethysmography. In 3.4 we have attempted to model and simulate the expected signal, both mathematically and Through data simulations. We have also explored the effect of different kinds of noise on the signal.

In the implementation chapter starting on page 29 we will give an explanation of the signal processing chain in 4.1 and a short explanation of the experiments conducted in 4.2.

The largest section is the results section starting on page 35. It can generally be divided into two themes relating to the two theme questions stated earlier. In sections 5.1 to 5.3 we will show statistical results from testing. The results here are largely there to answer the first main theme question. We will show results from tests of the effectiveness of different color spaces and use of the ICA algorithm. Section 5.4 to 5.8 show the results of experiments where we try to investigate different sides of non-contact photoplethysmography in an attempt to answer the second main theme question.

We have an expanded discussion chapter from page 71 where we will discuss the results found in the previous chapter. We will discuss the results in largely the same order as they are presented in the results section. We will also discuss

the future of a fictional non-contact photoplethysmography system based on the software we have developed.

In the conclusion starting on page 93 we will try to draw the results and discussion chapters together and try to answer the problem description.

In the appendix part A additional test data which there were not room for in the main report but can still hold useful information. In appendix part B there are software code samples. Technical specifications of the equipment used can be found in appendix C.

2 Theory

2.1 Human vital signs

Vital signs proposed obtainable from Photoplethysmography are heart rate (HR), blood pressure (BP) and respiration rate (RR).

2.1.1 Heart Rate

Heart Rate (HR) is defined as the number of beats of the heart per minutes (bpm) and are normally in the range of 40 to 200 bpm. HR is usually divided into two "operating modes" i.e. two ranges of values for HR that is considered normal[1] depending on the state which the body is currently in. The first range is known as the "resting pulse". This is the HR when the body has not been under physical strain for some time and is in a state of rest. In this state the HR is normally in the range of 60-100 bpm.[2] The other operating state is when the body is under physical strain. In this case normal range is from 60 bpm. to the maximum possible HR (HR_{Max}), where HR_{Max} is dependent on age, sex and physical fitness. Due to its many variables, it is difficult to make a general statement about what is considered a "normal" value for HR_{Max}[3] , but usually it is found in the area of 180±20 bpm.[32]

2.1.2 Blood pressure

Blood pressure (BP) is a measurement of the pressure exerted by circulating blood upon the walls of blood vessels. This, along with the heart rate, is one of the principal vital signs of the circulatory system. The BP is given in two measurement. The maximum pressure, called systolic pressure, is the maximum pressure exerted when the heart is in mid beat. The other pressure, called diastolic, is the general pressure when the heart is resting in between beats. The measurement of BP is given in millimeters of mercury (mmHg) and is given in a systolic/diastolic format. It is primarily the beating of the heart that is responsible for the BP. The elasticity of the blood vessels also play a role in between heartbeats, but it is mainly the heart beat we will focus on.

It is the difference in blood pressure which is responsible for the flow of blood from one part of the circulatory system to another. The rate of blood flow depends on the resistance to flow presented by the blood vessels. This means an obstruction or blockage in the circulatory system, often called an occlusion, will decrease the flow of blood through that region and lower BP on the distal side of the occlusion.

As with HR, it is hard to define a normal value, with large variations from person to person and, in any one person, from moment to moment. The national

[1] The term "normal" in this section must be viewed as a somewhat crude approximation. There are many individual variations between humans, and it is hard to define what a specific normal range of values are. The term normal here refers to a statistical average for a healthy adult.

[2] Children 1-10 years: 70-130 bmp.
Well-trained athletes: 40-60 bmp. [20]

[3] In a 2002 study [25], the formula for estimating HR_{Max} described as "least objectionable" was $HR_{Max} = 205.8 - (0.685 \times age)$

health services of UK has defined three ranges of blood pressure. Low (90/60 or lower), normal (between 90/60 and 130/80) and high (140/90 or higher) . The state of low blood pressure is known as hypotension and the state of high blood pressure is known as hypertension[31, 32].

2.1.3 ECG

ECG stands for electrocardiography and is graph a of the electrical activity of the heart over a period of time. It is one of the most useful human vital signs related to the cardiovascular activity of the human body. ECG is measured by placing electrical probes at certain positions on the body and then record the electrical activity caused by the beating of the heart. The ECG curve of a normal beating heart is a well known curve and can be viewed in figure 2.1.

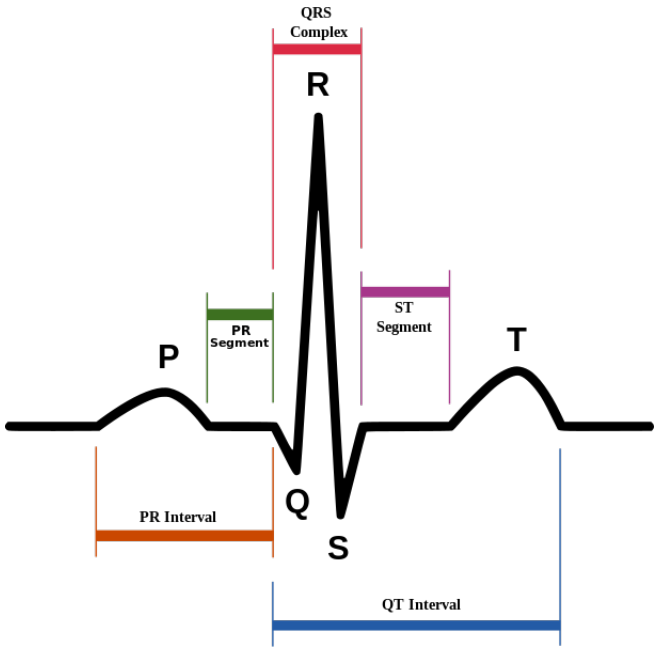


Figure 2.1 – ECG graph with the different waves and segments labeled [1].

The different waves and intervals in the ECG graph are divided into different parts and named with a single letter. It is divided into the P wave, the QRS complex and the T wave. The different intervals depict different activities performed by the heart.

First, in the P wave, electrical activity spreads from the SA node towards the AV node[4] and spreads from the the left to the right atrium. This makes the atria contract, pushing blood into the left and right ventricles. Next is the QRS complex. Between the P wave and QRS complex there is the PR interval which is measured from the start of the P to the start of the QRS complex. In this period the electrical impulse travels from the sinus node through the AV node

[4]These nodes (including the sinus node) are special regions of the heart that start or controls the electrical activity of the heart and sets the heart rhythm.

and enters the ventricles. The QRS complex is the rapid depolarization and contraction of the left and right ventricles. Due to the much larger muscle mass in the ventricles than in the atria, the QRS complex have a larger amplitude then the P wave. After the QRS complex and before the T wave is the ST segment. In this period the ventricles are depolarized. At the end comes the T wave which shows the repolarization of the ventricles. In figure 2.2 one can observe how the different phases of the ECG influences the blood pressure rhythm. It shows that when the QRS interval starts, the BP goes from diastole to systole.

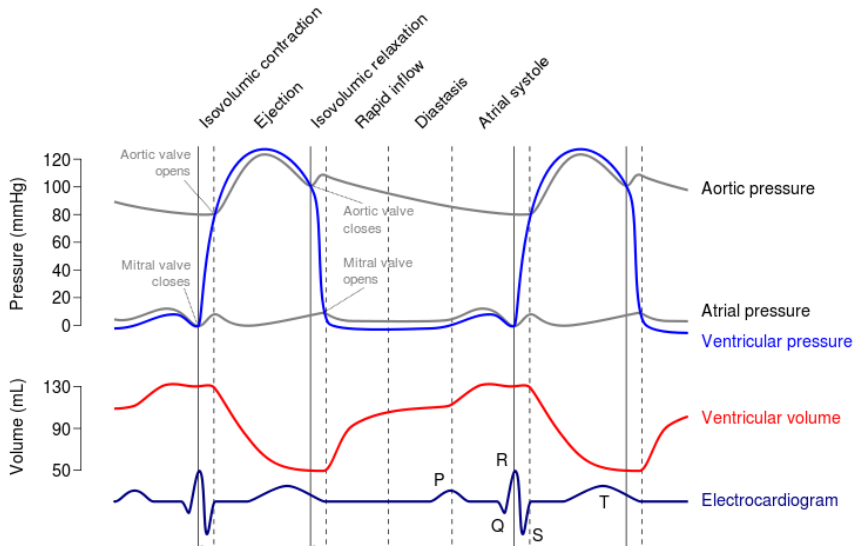


Figure 2.2 – Figure showing how the ECG interacts with the blood pressure[1].

2.1.4 Pulse transit time

Pulse transit time (PTT) refers to the time it takes a pulse wave to travel between two arterial sites. The PTT is proportional to the distance between the measurement sites and the speed which the arterial pressure wave propagates through the blood stream. The speed at which the arterial pressure wave propagates is directly proportional to blood pressure, this is due to an stiffening of arterial walls at higher BP [27]. With an rise in BP the pressure wave speed increases, and inversely when the BP decreases. This causes the PTT to be directly inversely proportional to the BP.

We can either calculate the PTT by recording the time interval between the passage of the arterial pulse wave at two consecutive sites, or we can use an ECG and time it with the R or Q wave, which corresponds with the opening of the aortic valve. If we set the P or Q wave from an ECG measurement as t=0, we can calculate PTT by measuring the time before we measure the maximum value PPG wave in a peripheral position[27].

Relationship between PTT and BP

Due to the direct relationship between PTT and BP, PTT has been brought

up as a non-invasive method of measuring BP. The mathematical relationship between BP and PTT was first proposed by Chen [8] and below is a summary of his derivation as shown in [18].

First, the velocity which the BP pulse propagates through the circulatory system, known as Pulse Wave Velocity (PWV), is described using the Moens-Kortwegg equation [19, 17].

$$PWV = \sqrt{\frac{gtE}{\rho d}} \tag{2.1}$$

E is the arterial elasticity, t is the arterial thickness, d is the diameter and ρ is the blood density. The constant g is gravitational constant, which for simplicity is usually omitted and the pressure is said to be hydrostatic. This equation is based on the assumption that, over shorter periods of time neither the elasticity, the thickness of the walls nor the diameter of the walls changes significantly.

The elasticity modulus E is said to increasing exponentially with increasing blood pressure so that:

$$E = E_0 e^{\gamma P} \tag{2.2}$$

E_0 is the elastic modulus at zero pressure, P is the blood pressure (mmHg) and γ is a coefficient ranging from 0.016 to 0.018 (mmHg^{-1}).

PTT is related to PWV by:

$$PWV = \frac{K}{PTT} \tag{2.3}$$

K is a proportional coefficient, indicating the distance the pulse has to travel between the two arterial positions. Using some manipulations[8] we get

$$P_e = P_b - \frac{2}{\gamma PTT_b}\Delta PTT \tag{2.4}$$

P_b is the base blood pressure level, PTT_b is the value of the PTT corresponding to the pressure P_b while ΔPTT is the change in the PTT.

This equation relates change in BP with change in PTT. This means that if one knows the preexisting BP and PTT, one can calculate the current BP based on change in PTT.

2.1.5 Microcirculation

Microcirculation refers to the flow of blood to and from the smallest blood vessels in the body. This is in contrast to the circulation to and from the organs which is called macrocirculation. The blood vessels on the arterial[5] side are called arterioles and are from 10-100 μm in diameter. The arterioles next transports the blood to the capillaries which are 5-8 μm in diameter. Back from the capillaries the blood flows to the venules which are 10-200 μm in diameter and next into the veins back to the heart. It is the capillaries which are the closest to the skin surface, which holds the blood we use to detect PPG signals with.

[5]Arterial means the blood vessels transporting fresh oxygenated blood *from* the heart to the rest of the body, while veins transports deoxygenated blood from the body *to* the heart.

The relationship between arterial blood pressure (ABP) and blood flow through the capillaries is complex, and ABP is not the only dependency. Circumstances such as room temperature, hormone level, osmotic pressure, distance from nearest artery and other unknown factors all influence the flow of blood through the capillaries, and more importantly from our perspective, the radiated PPG signal which we use for HR detection[26].

2.2 Non-contact photoplethysmography

Plethysmography stems from the Greek word plethysmos which means to *increase*. It is defined as *"[...] finding variations in the size of a part owing to variations in the amount of blood passing through or contained in the part."*[2].

In conventional plethysmography one measures volume changes in tissue related to the blood passing through it, e.g. with a strain gauge that measures changes in the circumference of an extremity. Arterial pulsations are the most significant reason for volume changes. Capillaries are largely non-compliant and will register only minor pulsations. The plethysmography is used as an indirect measure for ABP [15].

*Photo*plethysmography (PPG) means that one uses light to find information related to the plethysmography. There are two modes, *transmission*- and *reflectance*-mode PPG. Transmission mode measures how light is obstructed and absorbed by the tissue and uses a dedicated light source for this. For instance is pulse oximeters often a transmission mode PPG. It uses a LED or similar light source on one side of an extremity (earlobe, finger) and a detector on the other side. It detects the variability in absorption through the extremity due to plethysmography. Reflectance mode observes how the light is *reflected* from a light source, of the skin surface. The mode we will investigate in this paper is thus reflectance mode[28].

Non-contact refers to that there is no physical contact between the recording equipment used and the skin.

2.2.1 Hemoglobin absorption spectra

Human blood consists of 45 % red blood cells and 54.3 % plasma. The remaining percentage are other types of blood cells and proteins such as white blood cells and platelets[32]. The plasma is in itself mostly transparent, so the bulk of the reflectance and absorption we use to measure PPG signals are caused by the red blood cells circulation via microcirculation close to the skin surface. Therefore the absorption spectra of red blood cells is important to understand if one are to further understand the buildup of the PPG signal.

Red blood cells cytoplasm is rich with hemoglobin molecules, an iron carrying molecule which is used to transports oxygen through the body. It is this molecule which is responsible for the red color of the blood. In figure 2.3 which is taken from [35] and modified in [28], one can observe the absorption spectra of hemoglobin. The figure distinguishes between oxygenated hemoglobin (HbO_2) (1) carrying oxygen out to the body and deoxygenated hemoglobin (Hb) (2) returning from the body. The figure displays the absorption spectra in the spectra of visible light from 450-700 nm. One can observe that we have absorption peaks below 470 nm and between 525 and 590 nm, with a small dip for HbO_2 around 560-570 nm.

The values from [35] are obtained by transmission of light through *in vitro*[6] blood. Their transference to PPG signals which are acquired *in vivo* are supported by [12] where the reflectance of *in vivo* tissue are theoretically assessed and found similar to the *in vitro* case.

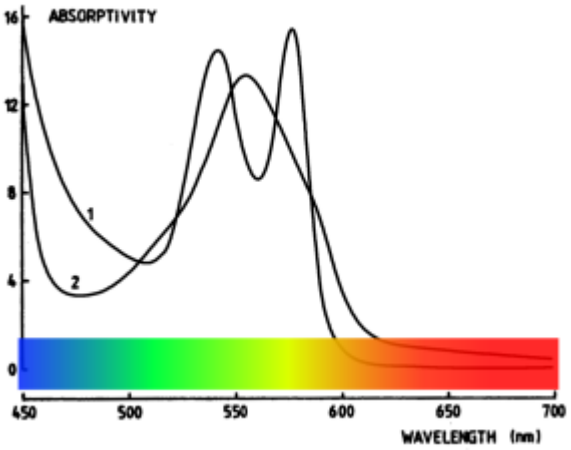


Figure 2.3 – Hemoglobin absorption spectrum. (1) HbO_2, (2) Hb. [L $\cdot$mmol$^{-1}\cdot$cm^{-1}]

2.2.2 PPG signal

The absorption spectrum observed in figure 2.3 shows one that light absorption is at its highest in the green color range, with some absorption in the lower blue. Since PPG uses the *changes* brought on by fluctuations in blood concentration, higher absorption means larger changes with these fluctuations. There will thus be a stronger signal in the frequency band with high absorption. In the RGB color space, the strongest variation should be observed in the Green channel, with some in the blue channel. Pixel values will be higher in the red channel, due to the reddish color of our skin, but little variation due to the flow of blood should be observed. These assumptions are corroborated by results in [23] and [28]. The HSL and HSV color spaces does not have these distinctions in to frequency ranges, so in which channels we will find the strongest signals are not yet known, and an attempt to discover this is done in this thesis.

The fluctuations of blood, and thus radiated PPG signal, is at least proportional to the ABP, even tho there is complicating factors not yet fully understood. This means that the PPG signals we receive is inversely proportional to the ABP. ABP is believed to be the main source of PPG signals, and the morphological similarity between the two signals support this [28, p.3]. It is suggested that the relationship between the PPG and the ABP is similar to that of traditional plethysmography and ABP [24, 3].

[6]Latin: *in glass*. Meaning studies conducted with samples taken from their natural environment i.e. blood removed from the body and experimented on in glass test tubes. This is in contrast with *in vivo*, which is a study on a sample in its natural environment i.e. blood still inside the body.

2.3 Color space

In this thesis, channels from specific color spaces are denoted with capital first letter while the color attributes they describe are written with normal lower case first letter.

A color space is a mathematical system for describing colors as a set of numbers,usually as group of three or four discrete numbers, known as the color *components*. A set of a specific component from a color space is known as a channel. The color space system enables one to describe the color, of for instance a pixel, with a discrete number of values. A color space is usually named after the first letter of the names of the components it contains. For instance the color space **RGB** is named after its three components **R**ed, **G**reen and **B**lue.

Each component is represented as a value within a certain possible value range, where the range is limited by the number of values possible by the number of bits used. The number of bits per component gives the number of possible discrete values possible for each channel, this is known as the color resolution. All modern color spaces have three components, with a possible fourth known as alpha, to describe transparency.

The most common implementation is the 24-bit implementation, with three components of 8 bits each. Higher resolutions with 16- and 32-bit representation exist, but are rarely encountered. Each components is usually described as a 8-bit unsigned[7] integer value. 8 bits give 256 discrete levels of color per channel in the range of [0-255] (0000 0000-1111 1111). For the 24-bit implementation this gives $256^3 \approx 16.7$ million discrete colors.

Vector representation

Since color space is represented by three values, it is common to represent color space as a three dimensional space, where each color has a position inside the space. The axes of the space are the color components. The intensity of each component is represented as the distance from origin. Figure 2.4 shows the RGB color space with Red, Green and Blue as the color axes. One can observe that the point of the color white is the point furthest away from the origin for all components.

2.3.1 Types of color space

In this section we will review the types of color space used in this paper, but first we must review some of the definitions used in color description as used in [13] and used by the ASTM[8]. Since color perception is a subjective measurement by humans, some of the definitions are also subjective and are defined as how something is *perceived* relative to another object.

Hue Can be viewed as the "pure" color, without other adjectives referring to the lightness and chroma of the color. When we refer to a color as red or blue, we usually refer to the hue.

[7] Unsigned integers does not support negative number. A signed 8-bit integer can range from [-127,128] while unsigned range from [0,255]

[8] ASTM International, formerly known as the American Society for Testing and Materials

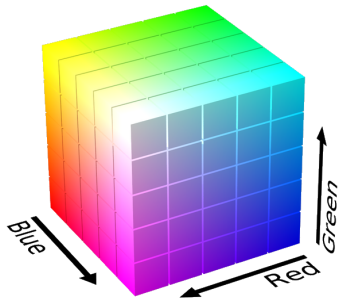


Figure 2.4 – RGB color space as a three dimensional space, each component representing an axis.
Source: Wikimedia commons

Luminance The radiance weighted by the perception of that wavelength by a human.

Brightness The attribute of a visual sensation according to which something appears to emit more or less light.

Lightness The brightness relative to the brightness of a similarly illuminated white.

Colorfulness The attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic.

Chroma The colorfulness relative to the brightness of a similarly illuminated white.

Saturation The colorfulness of a stimulus relative to its own brightness.

Brightness and colorfulness are absolute measures, which describe the spectral distribution of light entering the eye. Lightness and chroma are measured relative to some white point, it should therefore be more constant even when illumination of an object changes.

RGB

The RGB color space is the most common color space used for displaying images today. It is commonly used because it emulates how the human eye perceives color. The human eye has special cells to detect light in the red, green and blue frequency ranges. The brain later converts the magnitude of impulses from these three frequency regions and translate it to what we perceive as the color. Most display devices such as data screens and mobile phones try to emulate this setup. They employ a mixture of red, green and blue ordered placed in a matrix, then vary the intensity of each color so to simulate different colors.

RGB color space works on the same principle and describes what kinds of light needs to be emitted to produce a given color. It is therefor known as an additive color mixing, because is ads up the different component to create the wanted outcome. RGB is represented by the components red, green and blue

RGB $= \{R, G, B\}$ $R, G, B \in [0, 255]$. White is is a mixture of all colors at maximum intensity. In an 24-bit implementation, white is therefore represented as {255,255,255}. Black is the lack of any light and is represented as {0,0,0}. A completely green object would be represented as {0,255,0} and a completely red as {255,0,0} etc. RGB is easy to use when describing "simple" colors, but when describing different, more advanced characteristics such as the saturation and chroma of a color, it quickly becomes unintuitive. For instance the primary color red is described simply enough as {255,0,0}, but if we wish to describe different shades of red such as orange {255,115,115} or burgundy {132,15,15} we have to mix the different colors components in an unintuitive way. For this reason, in areas such as computer vision, one often employ other color spaces such as HSV where different shades of color can be described more intuitively.

RGB is represented as a Cartesian coordinate system in the shape as a cube as can be observed in figure 2.4. In all papers related to non-contact photoplethysmography, to our knowledge, only the RGB color space has been used represent color[29].

HSL/HSV

HSL stands for **H**ue, **S**aturation and **L**ightness while the **V** in **HSV** stands for **V**alue. These two color spaces are closely related and both are transformations of the RGB color space. While RGB has a cubic vector representation, HSL and HSV employs a cylindrical representation. In each cylinder, the angle around the cylinder represents Hue, while distance from the axis represents Saturation. In HSL the height represents Lightness while in HSV height represent Value. Saturation is defined differently in HSL and HSV.

Hue is the angular dimension, starting with red at 0/360°, green is at 120° and blue at 240°. Lightness and Value represent the gray color scale, so that black is at Lightness and Value 0, and white at Lightness and Value 1. In both HSL and HSV, the additive primary and secondary colors, that is red, yellow, green, cyan, blue and magenta, and the linear mixes between them, are arranged around the outside edge of the cylinder with Saturation 1. In HSV these have Value 1 while in HSL they have Lightness 0.5. In HSV, one mix with white to reduce Saturation. In HSV one mix with both white and black to reduce Saturation. The color spaces usually employ a 24-bit implementation. Hue is usually described as an angle between $H \in [0° , 360°)$ while Saturation and Value/Lightness is described as a percentage between 0-100 % $S, V, L \in [0, 1]$, **HSV** $= \{H, S, V\}$ and **HSL** $= \{H, S, L\}$. These values are transformed to a 8-bit representation during implementation. In figure 2.5 one can observe the geometry of HSL and HSV[29].

2.3.2 Difference between color spaces and advantages of HSL/HSV

The motivation behind using HSL/HSV is that it makes a more intuitive representation of different shades of colors than RGB. For instance the shades of red described in the RGB section are simply described with a Hue of 0°, with only variation Saturation and Value/Lightness. Orange is, in HSV, {0°,0.55,1} while burgundy is {0°,0.90,0.52}. From a computer vision perspective this makes it a lot easier to make algorithms based on color. For instance, if one wants to detect a red object in an image, one simply has to find all objects with Hue

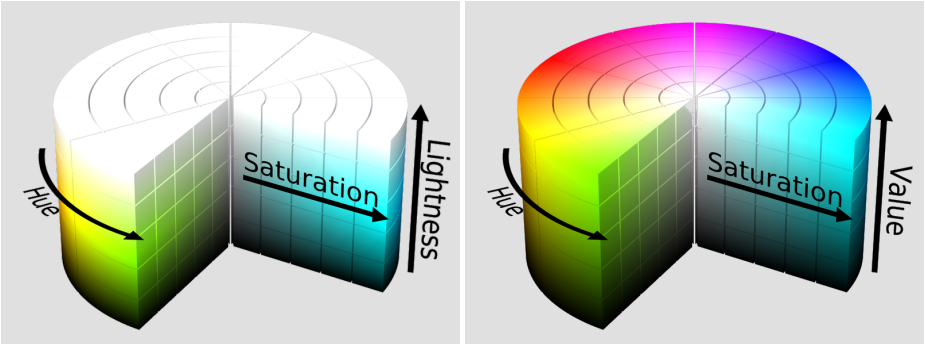


Figure 2.5 – (a)HSL and (b)HSV color space in cylinder geometry

of 0°. This makes it possible to make detection of color regardless of lighting conditions of that object, since this only affects Saturation and Value/Lightness values. If one were to implement such a system in RGB, one must take in to account all the different mixtures of components that could result in different shades of red.

With regards to PPG, HSL/HSV have one additional advantage. All the components of RGB color space are dependent on *both* the color of the object and the intensity of light reflecting from it. In HSV/HSL Hue and to a certain degree Saturation, are not dependent on this and *should* therefore be uncorrelated with changing lighting conditions. This will make a more stable recording, with channels able without a lot of the noise that is induced by the changing lighting conditions. At the same time there is a lot of the PPG information in both the changing color *and* absorption in the skin due to the flow of blood. How this affects the different component of HSL/HSV and how it relates to RGB is still not known and in this thesis we will therefore investigate this and evaluate if HSV/HSL offers any advantages to RGB.

Since the components of HSL/HSL are less correlated with the light intensity, they should be less statistically correlated with each other. This would improve the probability of extracting a clean signal of the HR using independent component analysis (ICA) on the different components. This will also be investigated in this thesis[11, 29].

2.3.3 Color transformation

In this thesis we will use the framework supplied by the Open Computer Vision (OpenCV) library. This is an open and commercially free library with a complete framework for handling graphics. It supports an extensive collection of functions for all kinds of image and video processing used in computer vision.

Since most images are handled in RGB color space, it is necessary to transform between color spaces. In openCV this is performed by an internal function called **cvtcolor.** In openCV the transformation between RGB and HSV is shown in equation 2.5 and the transform between RGB and HSL is shown in

equation 2.6.[22]

$$\begin{aligned} V &= max(R,G,B) \\ S &= \begin{cases} \frac{V-min(R,G,B)}{V} & if\, V \neq 0 \\ 0 & Otherwise \end{cases} \\ H &= \begin{cases} 60(G-B)/S & if\, V = R \\ 120 + 60(B-R)/S & if\, V = G \\ 240 + 60(R-G)/S & if\, V = B \end{cases} \end{aligned} \tag{2.5}$$

$$\begin{aligned} V_{max} &= max(R,G,B) \\ V_{min} &= min(R,G,B) \\ L &= \frac{V_{max} + V_{min}}{2} \\ S &= \begin{cases} \frac{V_{max}-V_{min})}{2-(V_{max}+V_{min}} & if\, L < 0.5 \\ \frac{V_{max}-V_{min}}{2-(V_{max}-V_{min})} & if\, L \geq 0.5 \end{cases} \\ H &= \begin{cases} 60(G-B)/S & if\, V_{max} = R \\ 120 + 60(B-R)/S & if\, V_{max} = G \\ 240 + 60(R-G)/S & if\, V_{max} = B \end{cases} \end{aligned} \tag{2.6}$$

2.4 ICA

ICA stands for Independent Component Analysis and is a special case of the wider concept of blind source separation (BSS). BSS is the concept that one can separate a set of source signal from a set of mixed signals, with very little or no information of the underlying source signals. ICA uses the statistical independence of non-Gaussian source signals to separate the different underlying components. A solution is attempted where the source signal, s, can be extracted from the original mixed channels, x, via a linear transformation, i.e. on on the form

$$x = \mathbf{M} \cdot s$$

or for a system of three sources and three mixed channels

$$[s_1(t), s_2(t), s_3(t)] = \mathbf{M^{-1}} \cdot [x_1(t), x_2(t), x_3(t)]$$

M is found by the ICA algorithm and is a n×n square matrix where n equals the number of recordings[9].

ICA is exemplified by its use in the "cocktail party problem", in which there are several people speaking at the same time in a room and several microphones records different mixtures of the speech signals. If the number of recordings is equal or greater than the number of sources, ICA can recover and separate the different speech sources (people speaking) based on the statistical independence that comes from their physical independence.

One of the limitations of ICA is that it cannot separate a greater number of sources than there are recordings, i.e. if there are four people talking at the cocktail party and only three recordings, one cannot separate all the sources from each other.

2.4.1 JADE

JADE stands for **J**oint **A**pproximate **D**iagonalization of **E**igenmatrices and is an matlab implementation of ICA. It is freely available online [6] and is an implementation by the original author of the algorithm[5]. It works by inputting the original signals as matlab row vectors. It then analyzes the recordings and outputs the so called **M** matrix which one can matrix multiply with to extract the original sources.

2.4.2 ICA on non-contact PPG measurements

ICA has been used with success in the papers by Poh [23] where it improved on the pulse rate measurements. However in the master thesis by Rustand[28], ICA did little to improve the results on video of higher quality, and any improvement was reserved for the lower quality video. Rustand found little overall use for ICA. These results stands at a contrast with each other and we will further investigate the matter in this thesis.

There are two limitations to ICA that have been omitted in previous papers using ICA on video recordings. The first is the assumption that there is an equal number of underlying sources and recordings. The underlying sources are signals caused by the HR and RR. With the noise factors and the biological nature of our measurements, there is an unknown number of signal sources on top of the two biological ones. It must therefore be assumed that the number of signal sources are greater then the number of recordings. This means that the signal sources we wish to record (HR and RR) can not be separated perfectly in any recording.

The other limitation is the assumption that the recordings used are statistically independent. The current method of acquiring different recordings is to split video in the RGB color space into its different channels and then see each channel as a separate recording, thus getting three recordings from one video source. But due to each channel in the RGB color space being dependent on *both* the color of a object and the intensity of light being reflected of it, we must expect a lot of correlation with changing lighting conditions on all channels. We can thus not expect the channels to be statistically independent of each other.

There is little to be done about the first limitation except removing as many noise and unwanted signal sources as possible. The only solution to this limitation is to increase the number of recordings. Splitting the video source into more channels will only degrade the signal. The only option is to increase the number of video recordings with more cameras. Due to the cost increase and added complexity this induces, this is not a viable solution in most cases.

The other limitation can be addressed by using a different color space than RBG. The channels of HSL/HSV are in theory[9] not all dependent on *both* color and light intensity. The channel should therefore be more statistically independent and therefore produce a better result with the ICA algorithm.

In a paper by Tsouri[30] a derivative of ICA, cICA (constrained ICA) has been used with success, but due to there not being an open implementation of this algorithm available, it has not been been investigated further in this thesis.

[9] Write "in theory" here because there is no perfect transformation between color spaces, and there will always be a bit of "leakage" between channels, meaning that input into one channel will influence the others.

2.5 Tracking algorithms and techniques

In this thesis we have used two algorithm for tracking and recognition. In this sections we will describe these two algorithms and how they work.

2.5.1 Face Recognition

The programming framework used for this thesis, OpenCV, implements support for face detection algorithms using Haar-like features. Haar-like features are digital image features used in object recognition and owe their name to their intuitive similarity with Haar wavelets[14]. Viola and Jones [34] adapted the idea of using Haar wavelets and developed the so called Haar-like features for use in object detection.

Haar-like features considers adjacent regions at specific locations in a detection window and sums up the pixel intensities in each region using integral images[10]. It then calculates the difference between adjacent rectangular regions. This difference is used to categorize subsections of an image.

For instance is the eyes of a human darker then the adjacent cheeks. Therefore a common Haar-like feature for face detection is two sets of adjacent rectangles that lie above the eye and the cheek region. One such detection is quite weak and only slightly better than random guessing, so a large number of different Haar-like features, or classifiers, are needed to distinguish and separate different objects and non-objects. In the Viola-Jones algorithm such Haar-like features are organized as a so called classifier cascade.

This means that Haar-like detection algorithm uses a large number of common contrast patters, or classifier cascades, of a object to detect it. This contrast information for the object one wish to detect is loaded in to the algorithm via data stored in XML files. The algorithm searches Through an image looking for the similar contrast patterns by separating an image into rectangular sub regions and looking for the contrast patterns locally. If a pattern is not found the algorithm moves on to the next sub region. It will go Through each picture multiple times looking for objects of different size. Calculation time is inversely proportional to the exponential of the minimum size of the sub regions. This minimum size is specified to the detection algorithm at calling [34, 22].

Due to this division into rectangular sub regions during detection, it will not detect faces tilted sideways or in profile. The algorithm is also relatively resource intensive and uses a lot of processing time, which could threaten the real-time demand of a system.

The Haar-like like feature detection algorithm is not limited to detection of faces. It can also detect other objects and human features like eyes and hands. This was attempted during work on the thesis but due to these features small size, long computational time and lack of regular and consistent detection was deemed to inefficient for use. But with further refinement it could potentially be used as a tracking algorithm.

[10]Integral images is an more efficient and quicker method of summing values in rectangular subsets of a grid. Introduced in 1984 by Franklin Crow[10].

2.5.2 SURF

SURF stands for **S**peeded **U**p **R**obust **F**eatures and is an algorithm for detection of strong features in digital pictures. It was presented by Herbert Bay et al. in 2006 [4]. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images. It is a fast and efficient way of detecting strong features in an image. These strong features can be contrasts on a surface, corners or sharp edges of an object. Basically they are features which a program recognizes easily. These features can be recognized from image to image, and can be used to track movement in between consecutive frames in a video.

3 Preliminary

In this chapter there will be some preliminary exploration into themes related to non-contact PPG. There is first a review of previous works, then a description of the video quality measurements we have and some general description of different video quality parameters and how they effect results. Last are modelings and simulations of the expected signal.

3.1 Previous works

3.1.1 Study by Verkruysse, Svaasand, Nelson

A survey of non contact ambient-light PPG was done in a 2008 article by Verkruysse, Svaasand and Nelson [33]. It showed that a PPG HR signal could be collected from the human face by using consumer grade video equipment and only ambient lighting.

Video of different subjects were recorded at either 640×480 or 320×240 resolution in the RGB color space while the subjects were sitting or laying down to reduce movement. ROI was placed manually and pixel values within the ROI was averaged over the Red, Green and Blue channel. They were able to detect PPG HR signals, with the best quality found in the Green channel. They also found that the best placement, with the strongest PPG HR signals relative to noise, was a ROI placed on the forehead above the eyes and below the hairline. This is shown in figure 3.1 where the optimal ROI placement is referred to as I in the figure.

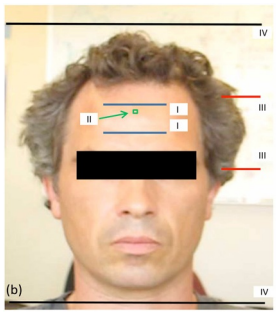


Figure 3.1 – Figure borrowed from Verkruysse, Svaaasand, Nelson article, showing different ROI placements. The best placement was found in the region marked as I in the figure.

3.1.2 Study by Poh, McDuff, Picard

In a 2010 MIT article, called "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation" [23] Poh, McDuff and Picard explores the idea of using non-contact PPG to automatically measure HR using only consumer grade equipment. They used an Apple iSight webcam, recording at a 640×480 resolution in a 24-bit RGB color space, to measure PPG HR signals from subjects. Their experiments were performed with varying amounts of sunlight as the only source of light. Tracking and motion

compensation was done using face recognition software. The subjects filmed was instructed to move around as if they were interacting with a computer normally. The ROIs were selected and channels averaged in a similar manner to [33]. In addition they used the ICA algorithm with the JADE implementation to separate the underlying sources. The measured PPG HR signals, both with and without using ICA, were compared to heart rate measurements from a blood volume pulse sensor. They also used statistical analysis of previous measurements to remove false positives from their measurements and to select the correct frequency peaks as the HR frequency.

Their results gave an overall root mean square error of 19.36 bmp for video measurements not using ICA and 4.63 bpm. when using ICA. This gives a strong cue that ICA can be used to improve PPG HR measurements in video.

3.1.3 Master thesis by Åsmund Rustand

Rustand wrote in 2012 a master thesis on the subject of Non-contact PPG [28] at NTNU. This thesis is a continuation of this work.

Rustand also found that one could acquire a reasonable quality PPG signal using consumer grade video equipment. The best signal was found on the forehead of the subject when the subject was not moving. As long as the subject was not moving, a signal of reasonable quality could be expected. A scheme to track and compensate for movement was used, by using face recognition software to detect the position of the face in each frame. The results during movement improved, but were vulnerable to the nature of the tracker. A filtering of the face coordinates alleviated some of these problems.

Rustand further found that ICA algorithm did not help unless the video was of very low quality, and that in some cases it actually deteriorated the signal.

In the best case scenario, with the subject not moving and video of high quality, heart rate estimates with an as low as 0.2 bpm average error, 0.6 bpm. maximum error and 0.2 bpm. standard deviation were found. With slight movement the numbers were 0.1, 2.1 and 0.7 bpm. respectively.

Further it was discovered that phase information is still present in the PPG signal and that it could be used to acquire further medical information from the subject such as pulse transit time.

Rustand also showed that compressed formats such as H.264 are to be avoided because high compression has a tendency to remove the color information we need.

Because of this the wmv video codec in a .avi container has been used in this thesis. This seems to be a reasonable compromise between compression, file size and availability of software and hardware support.

3.2 Video samples

All video used in testing the system has been divided into four categories. These categories is referred to as: ideal case, less ideal case, realistic case and worst case. These are in order of decreasing quality of the video. These quality measurements do not exclusively refer to standard video quality measurements such as bitrate, frame rate and compression, but refers to other assesments like mechanical movement of the subject, lighting conditions and background noise. This is an overall quality assessment related to how easy we believe it to

extract a PPG HR signal from the sample. We have implemented these quality assessments to standardize testing and to simplify the presentation of results. The four cases have these characteristics:

Ideal-case: The best possible conditions for measuring HR. No tracking required. Direct lightning of the subject forehead or arms. None to minimal mechanical movement of the subject and camera. Neutral, plain background with no movement. High quality video with a minimum of 720×1080 resolution, low compression.

Less-ideal: Still good conditions. Some tracking of subject might be required. Direct or indirect lighting of the subjects forehead or arms. Some mechanical movement of either camera or subject. Object in background but no movement. High to standard definition video, with minimum of 320×640 resolution, low compression.

Realistic-case: The conditions we would expect from a normal user sitting in front of a computer with webcam or a cellphone camera. Only ambient lightning, usually indirect. Some movement in front of the camera, both back and forth and side to side with reference to the camera. Tracking required. Object and movement in the background. Both high and standard definition quality video, varying resolution in the range of 720×1080 to 320×640. Low, medium and high compression.

Worst-case: Worst thinkable case where it might still be possible to acquire a signal. Backlight conditions that obscures the subjects skin and face. Large, fast movements. Tracking required. Subjects face could also be partially blocked by other objects.

3.3 Video parameters

In developing a functional system for HR detection using video, there are a number of parameters that must be set and evaluated against each other to create the best possible system. These are parameters like the number of frames per second (FPS), minimum resolution of each frame and number of subsequent frames required to find the HR frequency (window size).

3.3.1 FPS

An absolute minimum requirement for FPS is that it satisfies the Nyquist limit. The Nyquist limit states that the sampling frequency must be at least twice as high as the topmost frequency we are trying to measure, or $Fs \geq 2 * f_{Max}$.

For our system the maximum frequency is the maximum human HR, HR_{Max}. This is an individual value for all humans, but for this thesis it is set to 200 bpm, or 3.3 Hz. This gives a absolute minimum FPS of 2×3.3 Hz ≈ 7 FPS for our system.

3.3.2 Frequency resolution

A discrete fourier transfer (FFT[11]) of a signal divides the frequency spectrum of the input signal into N number of discrete frequency blocks equal to the number of samples in the input signal. The frequency spectrum has a range from $-Fs/2$ to $Fs/2$, divided into N blocks each with a range of $\Delta f = Fs/N$, the narrowness of Δf, i.e. the smallest frequency differences we are able to differentiate between is known as the frequency resolution. To increase the frequency resolution one either has to increase the number of samples or decrease sampling frequency. For a video system it is not always possible to vary the sampling frequency because this is often predetermined to a standardized video format in the range of 15-50 FPS. It is easier to increase resolution by increasing the number of frames in each sample, referred to as the window size.

We need at least a sufficiently high resolution to be able to separate the HR components from the lower frequency (<0.66 Hz) components usually found in the most samples. It is also desirable to get a resolution of at least ±3 bpm. (0.1 Hz) and usually higher. In this thesis we have set the minimum frequency resolution to ±1.5 bpm. (0.05 Hz). Increasing the number of samples has other negative consequences, mainly that it requires longer, stable, video shots. This longer video shot increases the risk of noise. This either due to mechanical movement of the subject (it is harder for a subject to be perfectly still for a extended periods of time) or light fluctuations in the surroundings. The HR might also fluctuate more over longer periods of time.

These negative consequences necessitates a compromise between frequency resolution and increased probability of noise caused by longer video samples. In this thesis we will investigate what the optimal trade off point is. Table 3.1 shows frequency resolution as a function of window size and FPS. One must note that one can have a larger window size at a higher FPS because it is not the number of samples in itself that increases noise, but the length of the temporal window used to capture the samples.

Window size	10 FPS	15 FPS	25 FPS	30 FPS	50 FPS	60 FPS
100	0.1	0.15	0.25	0.3	0.5	0.6
200	0.05	0.075	0.125	0.15	0.25	0.3
300	0.033	0.05	0.083	0.1	0.17	0.2
400	0.025	0.037	0.062	0.075	0.12	0.15
500	0.02	0.03	0.05	0.06	0.1	0.12
600	0.017	0.025	0.042	0.05	0.08	0.1
700	0.014	0.02	0.03	0.04	0.07	0.08
1000	0.01	0.015	0.025	0.03	0.05	0.06

Table 3.1 – Frequency resolution Δf [Hz] as a function of window size and FPS

One can observe from table 3.1 that the minimum window size required for the most used FPS settings in this thesis: 25, 30 and 50 FPS requires a window

[11] Modern discrete fourier computer algorithms are usually referred to as Fast Fourier Transfers, or FFTs

size of 500, 600 and 1000 samples to get an acceptable approximation of HR frequency.

3.4 Modeling signal

The signal we are trying to detect in this thesis is the Photoplehysmography (PPG) signal. In this section we will attempt to model the signal and classify its different noise components. It is modeled in similar way in [7] but without the use of an external light source. We will also emphasize more on the noise sources.

The signal we are attempting to detect is the fluctuations in color tone and light absorption in the skin due to the heart beat. Simplified, this fluctuation occurs because the capillaries are either increasing in size or have increased blood flow. We state simplified because the capillaries are a part of the complicated system of microcirculation described in theory chapter 2.1.5. But for this signal modeling this simplifications will suffice. The raw signal the photo sensor in the camera detects is denoted as x and is modeled as.

$$x = DC + AC + \sigma \tag{3.1}$$

Where the DC factor is the baseline value which the AC component, holding the HR and RR signal, oscillates around, σ is the noise component. The DC factor is the baseline color tone and absorption in the skin, while AC is the oscillation in skin tone and absorption due to the HR. In this thesis it is this component we are trying to isolate from the other component and detect. The DC component is actually time varying and is relative to temperature and other factors on the human body, but in the short time frames used to detect HR in this thesis we can assume that it stays constant. The AC component can be modeled as number of different biological factors as.

$$AC = S_{HR} + S_{RR} + \alpha \tag{3.2}$$

Where S_{HR} is the signal component due to the beating of the heart and S_{RR} is the component due to the respiration. α denotes all signal component due to other biological factors. It is usually indistinguishable from noise and miniscule compared to the other factors. Therefore we ignore this factor in the further modeling of the signal. S_{HR} and S_{RR} can both be modeled as time varying sine function relative to the heart rate and respiratory rate as follows.

$$S_{HR}(t) = A_{HR}(t) * sin(\frac{HR(t)}{60} \times t + \phi_{HR}) \tag{3.3}$$

$$S_{RR}(t) = A_{RR}(t) * sin(\frac{RR(t)}{60} \times t + \phi_{RR}) \tag{3.4}$$

The amplitude scalar A_{HR} is assumed to be at least correlated with the blood pressure, but for the short recordings we use here we assume that it is constant. ϕ_{HR} is the phase difference between the heart beats and when we detect the change in the skin. This is a function of the pulse propagation speed and the position of the measurement point. HR(t) is the instant heart rate at time t and is given in bpm. S_{RR} is similar to S_{HR} but operates at a much lower frequency than S_{HR}. We also get over harmonic frequency components that

operates at $n \times \frac{HR}{60}, n = 2, 3, 4...$ frequencies. These have a lower amplitude that the base frequency.

The DC component is not hard to remove. This can be done via a simple highpass filter. It is the σ component which is the hardest to isolate from the AC and we expect will be the biggest challenge. The noise component can be divided into several components.

$$\sigma = \sigma_w + \sigma_g + \sigma_m + \sigma_{other} \tag{3.5}$$

Here σ_w is the noise component due to white noise, σ_g is gaussian noise, σ_m is noise due to mechanical movement during the recording whileσ_{other} denotes are all other unknown sources of noise.

The white noise component, σ_w, is characterized by a flat frequency spectrum, with equal noise power in all frequency range. It is known as the background noise of a measurement. Measuring equipment all have a component of white noise in their measurements and the level of this noise is sometimes referred to as the noise floor of a measurement. Due to its flat frequency spectrum it is very hard to filtrate out this component and the only way to avoid this is to have higher quality measuring equipment which has a lower noise floor.

The Gaussian noise component σ_g is noise which has a probability density function equal to that of the normal distribution. We expect this component to be present in our measurement, with a normal distribution around the DC value, due to the biological nature of what we are measuring on (the skin). Due to the complicated nature of microcirculation we do not expect all part of the ROI we are measuring on to vary at the exact frequency and in phase with HR and RR. We expect some part to be out of phase with the main wave and some part to vary with frequencies with a normal distribution around the HR and RR frequencies.

The mechanical movement noise component σ_m is caused by the mechanical movement of the subject and the camera during recordings. This noise component is characterized by sharp changes in the the background level (DC level) and in the time domain it can be approximated to a Heaviside function (but with nonzero rise time). The fourier transfer of a Heaviside is a sinc[12] function. The sharper the edge of the Heaviside (meaning increase in amplitude and decrease in rise time), the wider the sinc function becomes. This means that strong or rapid mechanical movement in an recording could mask the S_{HR} and S_{RR} signals completely. There is also a mechanical movement that is in sync with the frequencies of S_{HR} and S_{RR}, this caused by our bodies moving slightly with the heart beat and respiration. σ_m is expected to be the biggest problem in detecting PPG signals due to its random nature. For instance if the subject moves during recording, we get a sharp change in signal level that masks the other signal component we are trying to measure. Because of this noise component it is essential that the subject either sits completely still or that we implement good movement compensation algorithms that moves the ROI during recording and compensates for movement.

The last noise component σ_{other} is a collection of all other unknown noise components. It can be caused for instance by shifting lighting conditions during

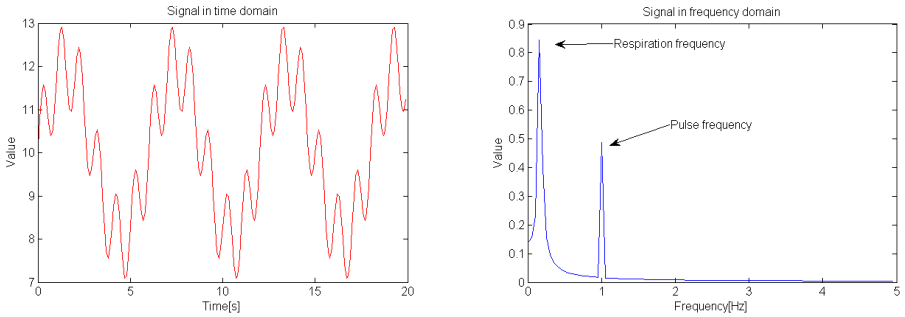
[12] A sinc function is defined as $sinc(x) = \frac{sin(\pi x)}{\pi x}$, and is a very common function within signal processing

recordings or other noise sources induces during compression and transfer of the video files.

3.4.1 Signal and noise simulations

In this subsection there are simulations of the signal and how it reacts to the different noise sources.

In figures 3.2-3.6 one can observe what the different types of noise does to the PPG signal in the time and frequency domain. In figure 3.2 one can observe what the signal should look like if there was no noise sources present. This is simply two sinus waves S_{RR} and S_{HR} oscillating around a base value (DC). In the frequency spectrum the two frequency peaks are easily visible, with a HR at 60 bpm. (1 Hz) and RR at 10 breaths per minute (0.1 Hz). For simplicity we have not included over harmonic frequencies as we would expect in a real case.



(a) Signal in time domain over 20 seconds

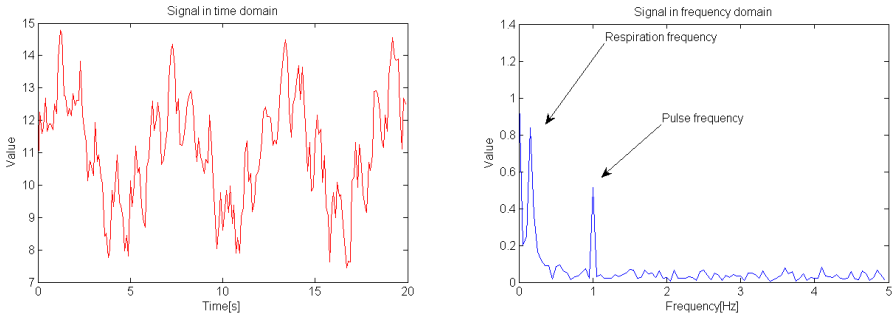
(b) Signal in frequency domain

Figure 3.2 – Signal with no noise sources. There are two sinus waves representing S_{RR} and S_{HR} varying with a frequency of 1 and 6 Hz and an amplitude of 1 and 2 respectively. The DC level set to 10. (a) shows the signal in the time domain over a period of 20 seconds while (b) shows the frequency domain with a sampling frequency of 10 Hz.

In figure 3.3 the σ_w noise component has been added on the original signal. One can observe that the two PPG signals are still easily visible as long as the white noise power does not become to large and one can acquire the HR and RR by a simple visual inspection of the time domain. One can also observe that the frequency spectrum is no longer flat in the higher frequencies.

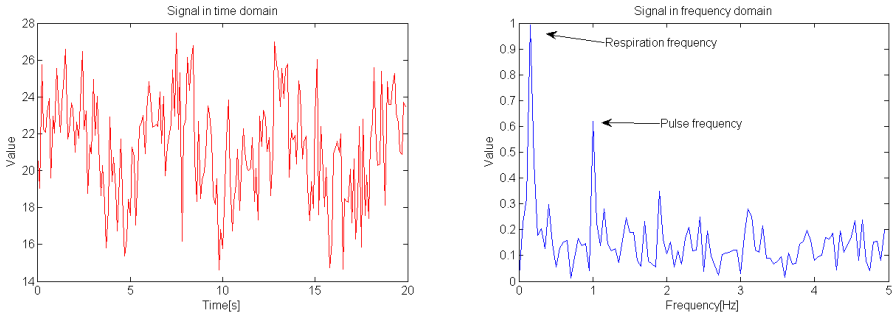
In figure 3.4 both the σ_g and σ_w component have been added. The Gaussian noise has a mean equal to the DC value and a standard deviation of 2. One can observe in figure 3.4(a) that it is now harder to visually observe the S_{HR} component in the temporal signal and we must refer to the frequency spectra to observe it. S_{RR} is still slightly visible.

In figure 3.5 we include an example of how σ_m would be expected to behave, without σ_w or σ_g being present for easier visualization. In this simulation we have simulated that the subject moves his head over a period of 1 second. This causes the DC value to change by 15 halfway through the simulation as



(a) Signal in time domain **(b)** Signal in frequency domain

Figure 3.3 – Signal with white noise sources. (a) shows the signal in the time domain over a period of 20 seconds while (b) shows the frequency domain with a sampling frequency of 10 Hz.

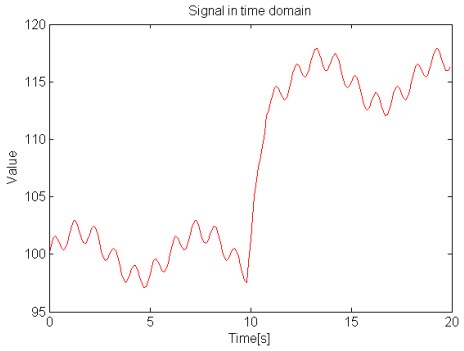


(a) Signal in time domain **(b)** Signal in frequency domain

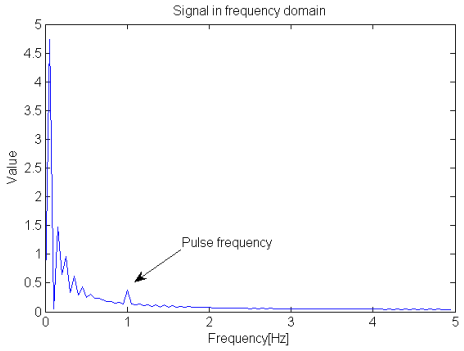
Figure 3.4 – Signal with white and Gaussian noise sources. (a) shows the signal in the time domain over a period of 20 seconds while (b) shows the frequency domain with a sampling frequency of 10 Hz.

is observed in figure 3.5(a). This is a very common occurrence in real world recordings and steep and sudden changes in DC values must be expected. One can observe in figure 3.5(b) the frequency spectra. One can observe the shape of a single sided sinc function at the lower frequencies. This sinc function obscures the frequency peak of the S_{RR} signal, thus making it impossible to single out this frequency peak and detecting a RR. The S_{HR} frequency peak is still observable because this resides at higher frequencies than S_{RR}, but its amplitude is substantially lower then the amplitude of the sinc function. Had the head movement been faster or had the DC value been shifted by more then 15, the S_{HR} frequency peak would have been obscured.

In figure 3.6 all sources of noise except σ_{other} have been added, using the same σ_m as in figure 3.5. One can observe that S_{RR} and S_{HR} are now completely obscured from visual inspection in the time domain and we have to rely on the frequency domain for detection. In the frequency domain S_{RR} is obscured by σ_m, while S_{HR} is still observable.



(a) Signal in time domain over 20 seconds. Note change in DC value halfway Through recording, simulating mechanical movement of the subject.



(b) Signal in frequency domain. Note single sided sinc function at the lower frequencies.

Figure 3.5 – Example of noise from σ_m. Simulating a subject moving his head after 10 seconds, over a period of 1 second. This causes the DC value to shift by 15 halfway Through the recording.

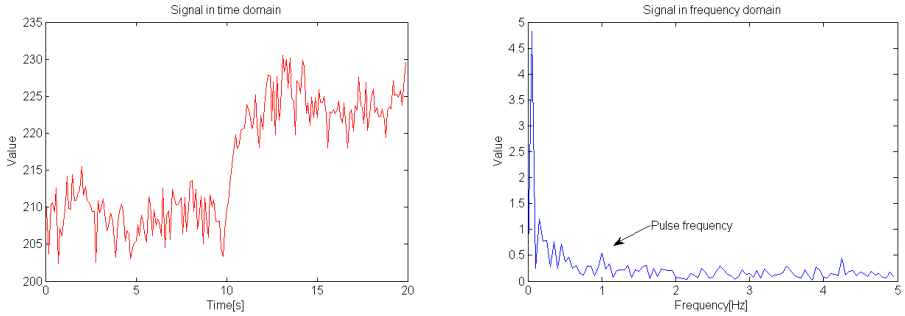
Figure 3.6 shows the signal most likely to be encounter in a real world setting, it also shows us what are the major challenges related to making a functional non-contact PPG system. These signal simulations show a couple of key factors we have to take in to account when designing this system, these are:

- Low frequency signals (<0.5 Hz) are likely to be obscured by noise sources. This means that the respiratory signal is most likely only to be detected in the most favorable of situations. It will also be hard to distinguish between a perceived respiratory signal and just a noise source.

- The noise source most likely to obscure the desired signals are due to mechanical movement, and is thus the primary challenge in obtaining a good non-contact PPG signal.

It shows that the most serious noise source is σ_m and making algorithms to combat this is an important part in making a functional non-contact PPG system based on video. σ_w and σ_g are noise sources that lie inherent in either the equipments we use for measurements or in the biological nature of what we are attempting to measure. So other than to have stable recording environments and to use suitable equipment, there is little one can do to avoid these noise sources. There are different coping mechanisms discussed to combat the effects of these noise source. For instance is source separation algorithms such as ICA used. These simulations have indicated that to observe S_{RR} is not probable under most circumstances, and that we will emphasize on the detection of S_{HR}.

To combat the effects of σ_m one can employ different strategies. The first strategy is to instruct the subject to sit completely still for the duration of a recording. This works well in an controlled environments, but in a real life setting this is not a viable solution. It is also hard even in an controlled environment to sit completely still for the entire duration of a recording long enough for a good PPG measurement ($\backsim 15 - 20$ s). The other strategy for combating



(a) Signal in time domain over 20 seconds

(b) Signal in frequency domain

Figure 3.6 – Signal with all noise sources except σ_{other}. Using the same σ_m noise as in figure 3.5, but now including σ_w and σ_g.

σ_m is to employ a tracking software or algorithm that detects movement and compensates by moving the ROI accordingly.

One can also use color spaces which have components that are more robust against sudden movements, such as HSL/HSV.

4 Implementation

4.1 Description of the Non-Contact PPG system

In the work with this thesis, an extensive framework in the C++ framework has been developed, mainly building on the existing OpenCV framework. In this section we will explain the signal chain and how the framework is built up.

Due to there not being an available ICA algorithm in C++, some Matlab programs have also been used, and all research using the JADE implementation of ICA has been done in Matlab. In this case all signal processing has been done in C++, and then the data dumped into Matlab through a script and the ICA processing done there. In all other regards the systems using ICA and not using ICA are identical.

4.1.1 Signal processing chain

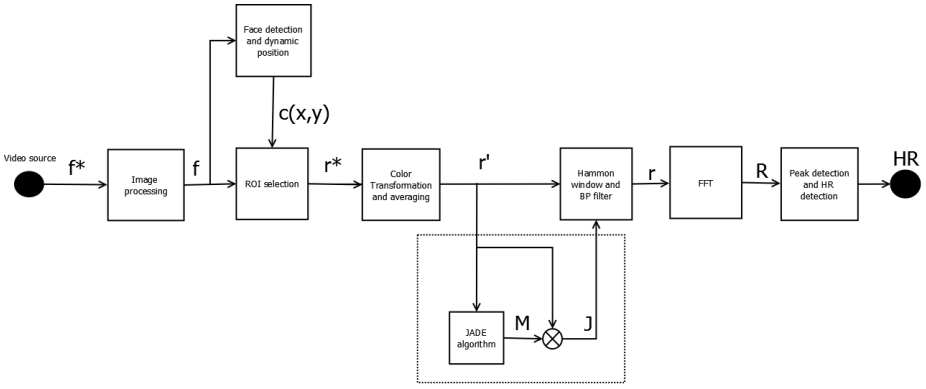


Figure 4.1 – Signal processing chain flow diagram

All steps in the signal process chain are shown in figure 4.1

Video source

Two video cameras have been used in recording samples for this thesis. The first is a Sony PMW-EX1 camera using a XDCAM[13] codec. Recordings were later converted to Wm format using Adobe Media Encoder. The other camera is a Microsoft Livecam webcam, recording directly to wmv format. All recordings were done only with ambient lighting.

From a video file, a subsample of video, starting at time t, with with N number of frames, equal to window size selected by user, was extracted. Each video frame is denoted as f*.

$$f_i^* = Videofile(t, t + N/Fs)\ , i = 1, 2...N$$

[13]XDCAM is a MPEG 2 derivative format

Initial signal processing

All signals were imported as wmv codec files in an RGB color space using built in functions in openCV, they were then converted into a C++ vector[14] of matrixes, with one matrix per frame. Some initial imaging processing was done on each frame to improve readings. The processed frames denoted as f.

$$f = IP(f^*)$$

ROI selection and tracking

A ROI[15] was selected. This was either done manually or automatically by the program, depending on user input. The manual mode consisted of placing a rectangular ROI manually, first selecting the ROI size over the desired region. Automatic mode consisted of letting the software detect faces in the picture and automatically place an ROI over the forehead. Both mode supported multiple ROIs, so that one video could support multiple ROIs, all taken at the same time so that one could compare them temporally. This was used for the phase difference experiments.

One could chose either dynamic tracking or static position modes for the ROI. With static mode the ROI stayed in place for the duration of the window. With dynamic tracking the program attempted to compensate for movement using position tracking algorithm which will be described later.

Image within ROI is denoted by r*. Windowsize is N and the number of ROIs taken from one frame is K. The coordinates for the ROI within the original frame, c(x,y), and the size of the ROI, s(w,l), are either given by the user or by the dynamic position algorithm.

$$\mathbf{r}^*_{i,k} = ROIselection(f_i, c_k(x,y), s(w,l)) \, , i = 1, 2...N, \, k = 1, 2...K$$

For each frame, the ROI of the frame was transferred to a new C++ vector, this was done for a windowsize number of frames and K number of ROIs. The original image in r* were then transformed into the two other color spaces HSL and HSV according to the transfer described in theory chapter 2.3.3. The average color in each channel was then calculated over each ROI, denoted as r' , where L is the total number of channels (9 in total, 3 for each color space).

$$\mathbf{r}'_{i,k,l} = CTA(\mathbf{r}^*_{i,k}) \, l = 1, 2...9$$

Dynamic position tracking and face recognition

In the automatic mode it was assumed that one attempted to obtain a PPG signal from the forehead of the subject. The cascade classifier face detection algorithm was employed and a ROI was placed automatically over the forehead. This position was automatically updated after a set number of frames (usually 150). In between each face detection update the feature tracking algorithm was used. Both algorithms are described in theory chapter 2.5.

[14] A C++ vector is known as a dynamic array of data, also known as a linked list. It enables us to make a consecutive list of any type of data, and in this case a consecutive list of images (frames).

[15] ROI stands for Region Of Interest and is a subimage within an image.

The dynamic positioning algorithm was used to compensate for any movement of the subject during recording and to reposition the ROI between frames accordingly. The algorithm used the SURF algorithm described in theory chapter 2.5.2 to detect traceable features in the picture. Using this feature between two frames it could calculate how much the feature had moved between frames and used this to correct the position of the ROI between subsequent frames. If a position p is a function of x and y-coordinate such that p(x,y) and p_n the current position of a feature, the algorithm takes the position of the same feature in the previous frame p_{n-1} and calculate the difference d between them.

$$d_n = p_n - p_{n-1}$$

The new coordinates for the ROI, c(x,y) is then

$$c_n(x, y) = c_{n-1}(x, y) + d$$

This updated position was transmitted to the ROI selection system for each frame.

Filtering and FFT

Each channel in r' had their mean value subtracted to remove DC. Then each of the vectors were multiplied with a hamming window with a length equal to N. Hamming windows are defined as:

$$HW(n) = 0.5(1 - cos(\frac{2\pi n}{N-1}))$$

This window function is used to reduce side lobes in the frequency spectrum.

Further the signals are bandpass filtered within the HR range (HR lies between 50-200 bpm.). The bandpass filtered channels are denoted r in the figure.

$$\begin{aligned} \{\mathbf{y}'_{i,k,l}\}_{i=1,2...N} &= \{\mathbf{r}'_{i,k,l} - mean(\mathbf{r}_{i,k,l})\}_{i=1,2...N} \\ \{\mathbf{y}_{i,k.l}\}_{i=1,2...N} &= \{\mathbf{y}'_{i,k,l} \cdot HW\}_{i=1,2...N} \\ \{\mathbf{r}_{i,k,l}\}_{i=1,2...N} &= \{\mathbf{y}_{i,k,l} * BP\}_{i=1,2...N} \end{aligned}$$

To find the frequency spectrum a 1 dimensional Fourier transform was done on each color channel. To find HR we took the peak frequency found in the Fourier transform within the HR range.

$$\begin{aligned} \mathbf{R} &= FFT\{\mathbf{r}\} \\ HR &= max(abs(\mathbf{R})) \end{aligned}$$

ICA Algorithms

When using ICA algorithms, the data in r' was dumped from the C++ program to a swapfile to be read by a Matlab program. The reason for this was that there was no implementation of ICA in C++ freely available. The transform therefore had to be performed in Matlab. The Matlab operations are shown as the dotted line in figure 4.1. The ICA algorithm works by separating the different sources as described in theory chapter 2.4.

The M matrix was acquired by inputting the three channels of each color space and outputting the M matrix.

$$\mathbf{M} = JADE\{\mathbf{r}'\}$$

The underlying physical signal matrix J were uncovered by

$$\mathbf{J} = \mathbf{M^{-1}} \cdot \mathbf{r}'$$

$$[j_1(n),\, j_2(n),\, j_3(n)] = \mathbf{M^{-1}} \cdot [r'_1(n),\, r'_2(n),\, r'_3(n)]$$

4.2 Experiments

All experiments performed in this thesis as described in results chapter 5.5 to 5.7 were performed at the intervention center at Rikshospitalet in Oslo during september 2012. They were performed under the supervision of Prof. Ilangko Balashingham. All experiments were performed on the author, unless otherwise specified.

Video was acquired using a Sony PMW-EX1 camera. ECG and pulse oximeter measurements were acquired using professional digital medical equipment present at Rikshospitalet. They were synchronized with the camera using a clock and timestamps in the video. Some physical input (whacking a sensor, giving a sharp pulse in the ECG, also visible in the video) was used to confirm synchronization.

The subject was sitting in a chair, except when the feet were filmed. In that case the subject was sitting on an operation table with his feet stretched towards the camera.

4.2.1 Statistical results

In previous works on non-contact PPG, statistical results were presented as average error, maximum error and standard deviation of error in bpm. when comparing non-contact PPG results with other HR measuring devices (usually a pulse oximeter). We find that this measurement can give a wrong representation of results. If for instance we have a set of five video samples and attempt to find HR using two different techniques. Imagine the first method gets a random noise peak on the first measurement, leading to an error of 100 bpm. from the pulse oximeter, but on the remaining measurement it finds the PPG HR signal perfectly. This gives an average error of 20 bpm.

On the other hand, the other technique fails to find the PPG HR in any of its samples. But due to the noise being high at the lower frequency ranges, just below where we would expect to find the HR, it detects the low frequency noise as the HR. This could give a result only 10 bpm. wrong each time, leading to an average error of just 10 bpm. Statistically, the second method appears superior even though it failed at finding the PPG HR signal, when only measuring average error.

To counter this we use a binomial statistical measurement instead, with a success/failure parameter instead of average error. We measure success as finding the HR signal within the frequency resolution of a given test, usually within ± 1.5 bpm. of the pulse oximeter value. Failure is measured as all results

falling outside this region. We believe this binomial measurement gives a truer representation of the efficiency of a given technique.

5 Results

This chapter is divided into two parts along the lines of the general themes described in the introduction. The results related to the first theme is covered in section 5.1 - 5.3 while results relating to the second theme is placed in sections 5.4 - 5.8.

Under the first theme is first sub chapter 5.1 - 5.2 that shows results regarding the use of ICA on non-PPG systems. The subsequent section 5.3 shows statistics on the efficiency of the different color spaces regarding HR detection.

Under the second theme is first sections 5.4 - 5.5 which shows results of experiments related to acquiring PPG signals from different positions and under different circumstances. Section 5.6 and 5.6.4 show results of phase difference experiments. In section 5.7 we show results regarding the synchronization of ECG and PPG signals. Section 5.8 show results of experiments done on laparoscopy and colonoscopy video.

All video samples in this section are referred to by the quality measurements given in preliminary section 3.2. In most results we have removed the Hue channel from HSV, this due to it being defined identically to HSL, and will therefore not yield different results than those found Hue from HSL.

Most graph values are the raw values given after filtration or transform. The values off FFTs are divided by the length of the sample. In some graphs the value of a signal is normalized between -1 and 1 for easier visualization, in those cases it noted in caption or in text.

5.1 Correlation between RGB and HSL signals

As described in section 2.3, the components of the HSL color space is expected to be less correlated than components of RGB color space, thus being more statistical independent and more susceptible to ICA transform. A comparison of different video clips were used to test the assumption that HSL channels were more independent. Due to the similarities between HSL and HSV, a test of correlation in both were deemed unnecessary. A ROI was extracted from each video frame and average color value over each ROI was calculated in RGB and HSL color space. This was done for 1000 consecutive frames to get the temporal change between frames. Each component was then exported to matlab and cross correlation coefficient between the different components were calculated using the **corrcoef** function in matlab. The results are shown in table 5.1. The higher the number, the greater the cross correlation between the channels is. A total of five video samples were used, all with varying quality to test different situations.

5.2 Improvement using ICA on video in HSL color space

In this section we will evaluate the use of ICA on video in HSL and HSV color space and compare it with ICA in RGB color space and determine if it offers any improvement in HR detection.

The statistical results in this section are measured as either a success or failure, i.e. as binomial value. This being either if the sample detected a HR frequency within a margin of error proportional to the frequency resolution of

Video sample	Red Green	Red Blue	Green Blue	Hue Lightness	Hue Saturation	Saturation Lightness
1	0.9969	0.9981	0.9986	0.3151	0.3226	0.9660
2	0.9974	0.9942	0.9928	0.4830	0.8002	0.7567
3	0.9950	0.9970	0.9979	0.0463	0.1089	0.9845
4	0.9884	0.9960	0.9970	0.1095	0.4587	0.6847
5	0.9782	0.9940	0.9934	0.2906	0.4317	0.5335
Average	0.9912	0.9959	0.9959	0.2489	0.4244	0.7851

Table 5.1 – Correlation between different color space component.

the sample or not. See implementation section 4.2 for reasoning behind use of this measurement technique.

5.2.1 Visual presentation of ICA on ideal case video in HSL color space

Having shown that the HSL color space is more statistical independent than regular RBG color space, ICA should be more efficient at separating the different components than using ICA on RGB color space. The trace shown in figure 5.2 were obtained using ICA on the trace shown in figure 5.1. In figure 5.3 one can observe the frequency spectrum of the same sample before and after ICA. In this figure we have also included the frequency spectrum of the Green channel from RGB taken from the same sample, with and without ICA. Two different ideal case videos were reviewed in a similar manner, giving similar results.

A statistical comparison of ideal case type video with ICA has already been performed in [28] and will not be performed here. We will in this thesis concentrate more on the effect of ICA on lower quality type video.

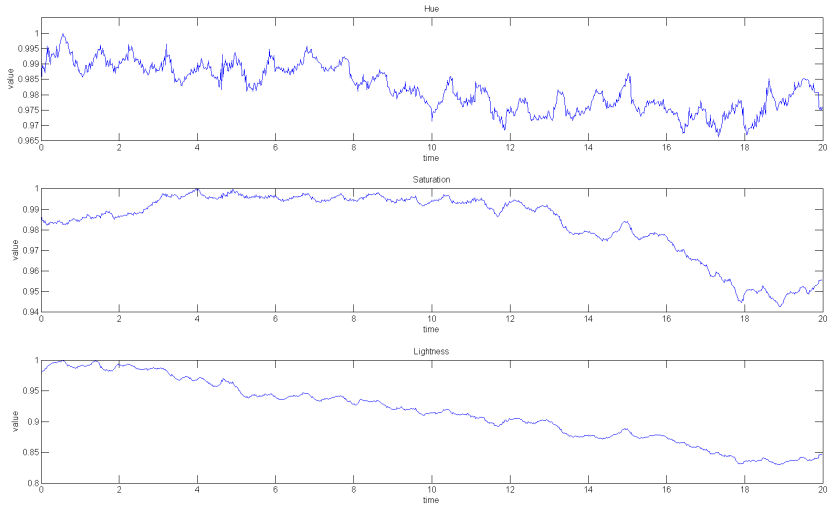


Figure 5.1 – Trace of HSL color space without ICA, ideal case video

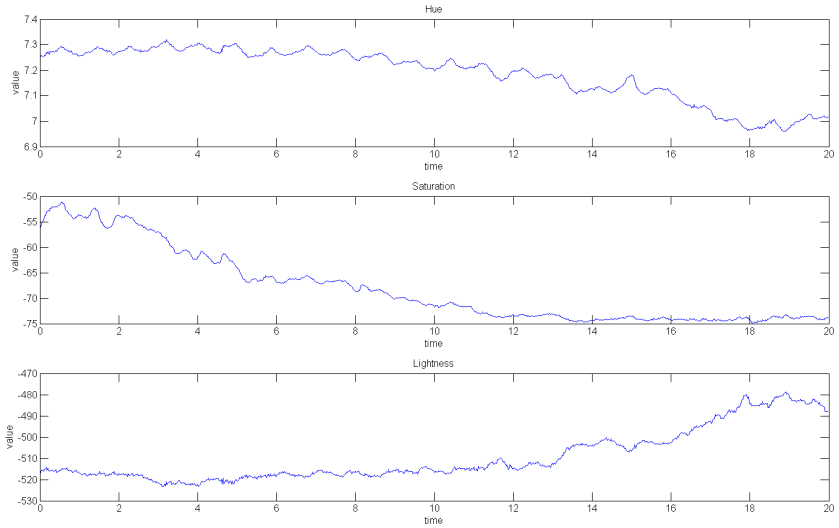


Figure 5.2 – Trace of HSL color space with ICA, ideal case video

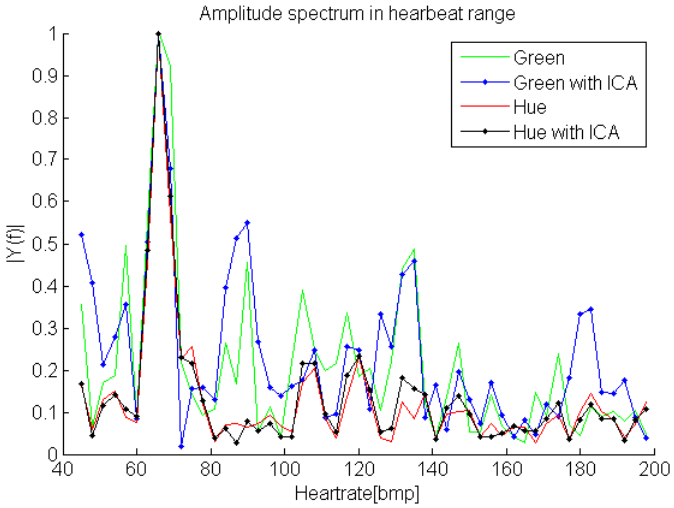


Figure 5.3 – FFT of Green channel, Green channel with ICA, Hue channel and Hue channel with ICA. Values normalized to 1. Ideal case video.

5.2.2 Statistical results of ICA on less ideal case video

ICA was tested on less ideal case video. A total of 44 samples were tested on in RGB and HSL color space, while 30 samples were tested on in HSV color space. The difference in number of samples for the different color spaces are due to an earlier assumptions that HSL and HSV would yield the same results and a test on HSV was deemed unnecessary. Later experiments indicated that this was an erroneous assumption so tests on HSV were performed, although with a reduced number of samples. Each sample were 500-1000 frames long at 25 FPS. Samples were obtained by using a Sony PMW-EX1 camera. Results can be observed in table 5.2.

Channel	Average success rate
RGB Green	0.59
RGB Green with ICA	0.57
HSL Hue	0.64
HSL Hue with ICA	0.52
HSL Saturation	0.53
HSL Saturation with ICA	0.63
HSL Lightness	0.26
HSL Lightness with ICA	0.50
HSV Hue	0.66
HSV Hue with ICA	0.43
HSV Saturation	0.53
HSV Saturation with ICA	0.63
HSV Value	0.50
HSV Value with ICA	0.60

Table 5.2 – Statistical results of ICA on less ideal case video. 44 samples for RGB and HSL, 30 for HSV.

5.2.3 Statistical results of ICA on realistic case

Realistic case videos were obtained using a Microsoft LifeCam webcamera. Samples were from 500-700 frames long at 30 FPS. Multiple subjects were used. 40 samples were taken for HSL and RGB, and 26 samples for HSV. Results can be observed in table 5.3.

5.2.4 Worst case

Attempts were made to read PPG signals in video in the worst case category. Unfortunately the movement in these videos were too excessive for the tracking software to be able to compensate. Only occasionally were it possible to get samples good enough to be able to extract a PPG signal. When a usable sample was obtained the results seemed to match those found in the realistic case, however the number of useable samples were too few to be able to draw any conclusions on.

Channel	Average success rate
RGB Green	0.20
RGB Green with ICA	0.56
HSL Hue	0.65
HSL Hue with ICA	0.60
HSL Saturation	0.65
HSL Saturation with ICA	0.55
HSL Lightness	0.15
HSL Lightness with ICA	0.55
HSV Hue	0.71
HSV Hue with ICA	0.59
HSV Saturation	0.70
HSV Saturation with ICA	0.35
HSV Value	0.04
HSV Value with ICA	0.54

Table 5.3 – Statistical results of ICA on realistic case video. 40 samples for RGB and HSL, 26 for HSV.

5.3 Performance of different color spaces

In this section we will evaluate the efficiency of the different color spaces and channels in a realistic situation. With a realistic situation we mean the video recording environment and circumstances most likely to be encountered by a potential mobile phone app or computer program using a webcam. The quality of the video could be compared to the lower end of the realistic case quality evaluation.

Multiple subjects were recorded using a standard Microsoft LifeCam webcam at 30 fps, with windowsize of 500-700 frames, giving a frequency resolution of 3.6 - 2.4 bpm. All camera settings were set manually before recording so they would not change during recording. The video background and circumstances of recording were similar to those to be expected if one were to record a video in a normal office situation. The background were non-neutral and had occasional movement. Figure 5.4 shows a typical recording setting. Only ambient lighting were used. During daytime the primary lighting source was indirect daylight through windows and during nighttime standard indoor lighting from fluorescent lighting fixtures was the primary source. Subjects were instructed not to move excessively, but in all other regards to move as they would during normal interaction with a computer. Tracking and detection software were employed to first detect the face, find its position and then to place an ROI over the forehead. Tracking software were used to move the ROI so to compensate for movement. The recognition software was run at intervals each 150 frames so as to compensate for drifting of the ROI.

During recording of a sample, if the tracking and detection software made a obvious mistake, the sample was discarded. For instance if the coffeemaker was mistaken for a face and the software tried to detect its pulse (it didn't have one), the sample was discarded from the experiment. Also, the tracking software had some problems with hair and would sometimes move the ROI from its intended position. The samples were discarded due to this being a test of the efficiency

of the different color spaces and not of the tracking and detection algorithms. Se discussion section 6.3.2 for evaluation and discussion of the tracking and detection algorithms.

Table 5.4 shows experiment results. A total of 95 samples were used. The HR of the subject was recorded with a pulse oximeter and compared to the PPG HR pulse detected by the software. If the PPG HR pulse was correct within the frequency resolution of the recording, the sample was regarded as a success.

Figure 5.4 – Standard recording setting for comparison of color space experiment. Noisy background and only ambient lighting. ROI placement is shown as a red rectangle in figure.

Channel	Average success rate
RGB Green	0.17
HSV Hue	0.46
HSV Saturation	0.57
HSV Value	0.07
HSL Saturation	0.53
HSL Lightness	0.10

Table 5.4 – Results from comparison of color space experiment. A total of 95 samples were used. Window size was between 500-700 frames. Samples acquired using a window LifeCam webcam under realistic conditions.

5.4 Detection of HR in feet

The experiment was designed to figure out if it were possible to observe a PPG HR signal in other parts of the body than the forehead and palm of the hand,

which to our knowledge has been used in all previous papers on non-contact PPG.

In the first experiment, efforts were made to determine if it was possible to detect PPG HR from the skin under the feet. The camera was pointed at the underside of the feet, while the subject was sitting on a table. The subject was connected to an ECG and a pulse oximeter to detect the HR. ROI was placed on the underside of the feet as show in figure 5.5. In figure 5.6 one can observe the frequency spectra of two runs with 1000 consecutive frames taken from a single video, taken from the Green channel in RGB color space. Two experiment with a similar setup to this were performed: One with and one without occlusions. In this section only the experiment without occlusion is shown. For the first run oximeter measured HR varying from 58, up to 67 and then down to 62 bpm. For the second run HR increased from 63 to 78 bpm.

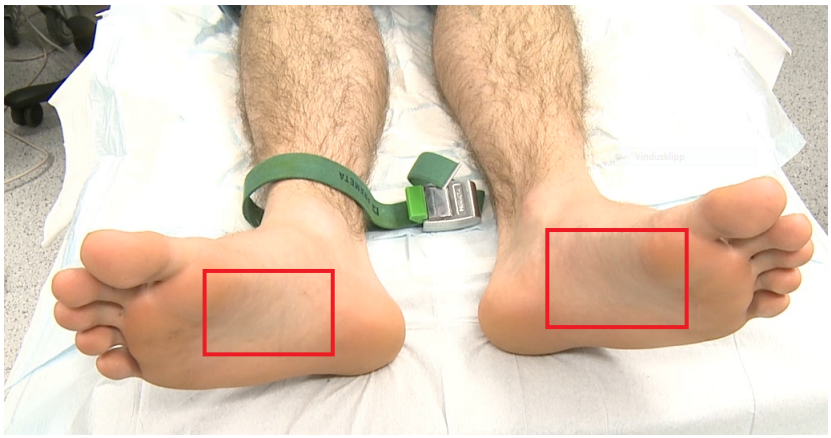
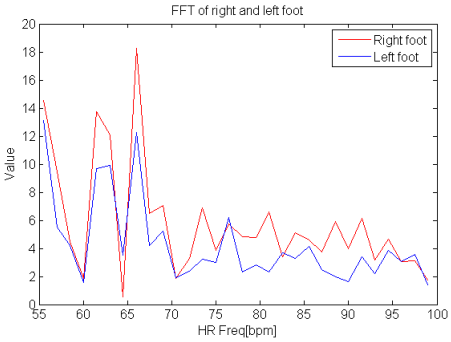
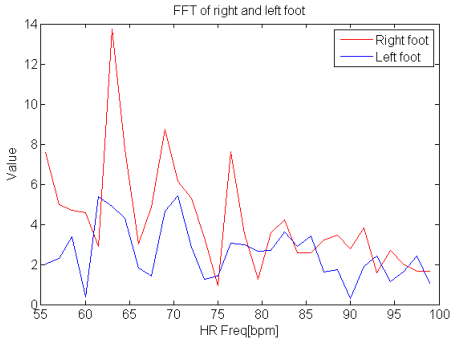


Figure 5.5 – Experiment setup and ROI placement for HR detection on underside of feet. ROI shown as red rectangles.



(a) FFT of video 1 (1000 frames). HR varying between 58 and 67.



(b) FFT of video 2 (1000 frames). HR increasing from 63 at start of video to 78 at end of video.

Figure 5.6 – FFT of video of underside of feet. (a) Video 1 (b) Video 2

The next experiment was similar to the previous. This experiment was designed to detect HR from the skin on top of the feet. See figure 5.7 for experiment setup with ROI placement.

A total of two individual videos were shot, each with a length of approx. 3 minutes. Each video was divided into 4 samples at 1000 frames each. HR was detected by oximeter. For video 1 the HR varied around 75 bpm and for video 2 the HR varied around 70 bpm. Observe figure 5.8 for frequency spectra from a representative sample, (a) is from video 1 and (b) is from video 2.

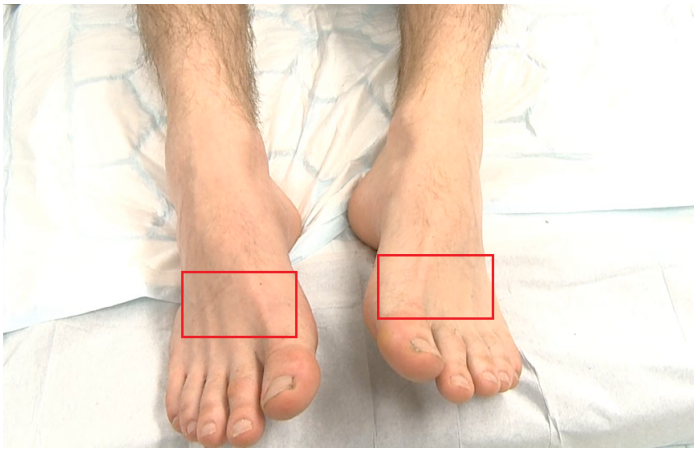
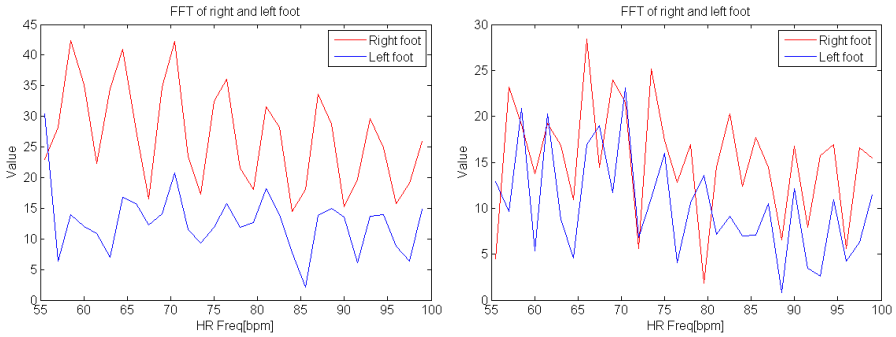


Figure 5.7 – Experiment setup and ROI placement for HR detection on overside of the feet. ROI shown as red rectangles.



(a) FFT of video 1 (1000 frames). HR varying around 75 bpm.

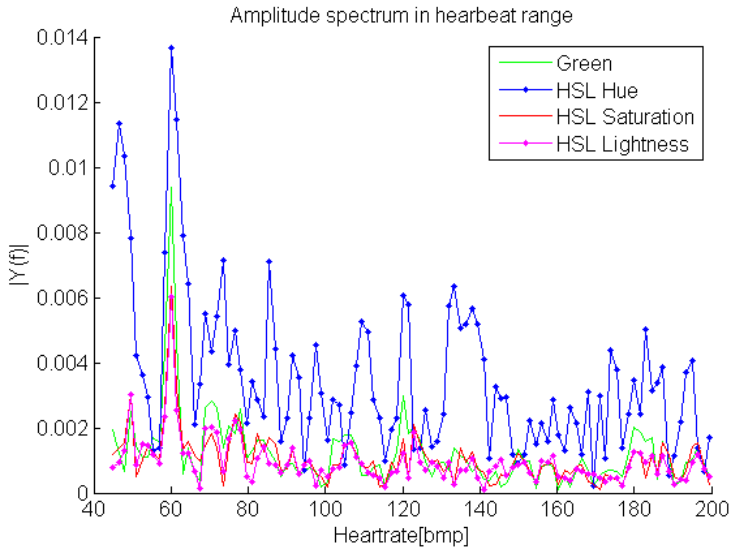
(b) FFT of video 1 (1000 frames). HR varying around 70 bpm.

Figure 5.8 – FFT of video of overside of feet. (a) Video 1. (b) Video 2.

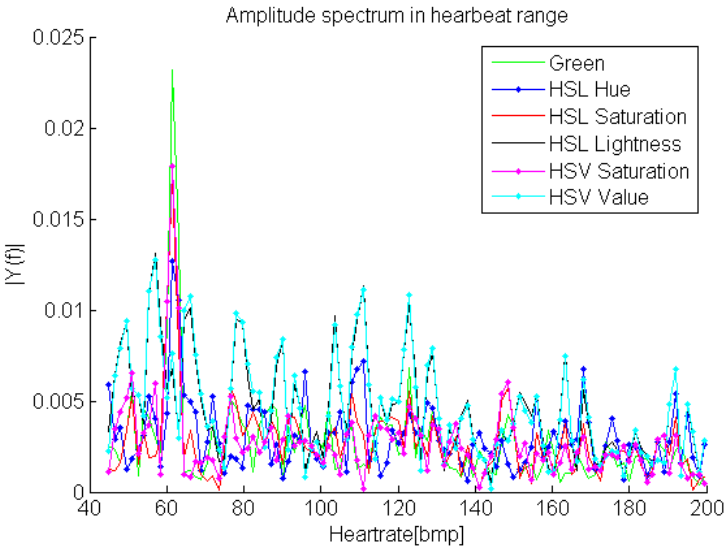
Figure 5.9 – Setup for experiment using PPG to detect HR in darker skin tones. ROI is shown as the red rectangle on the forehead of the subject.

5.5 HR detection with darker skin tones

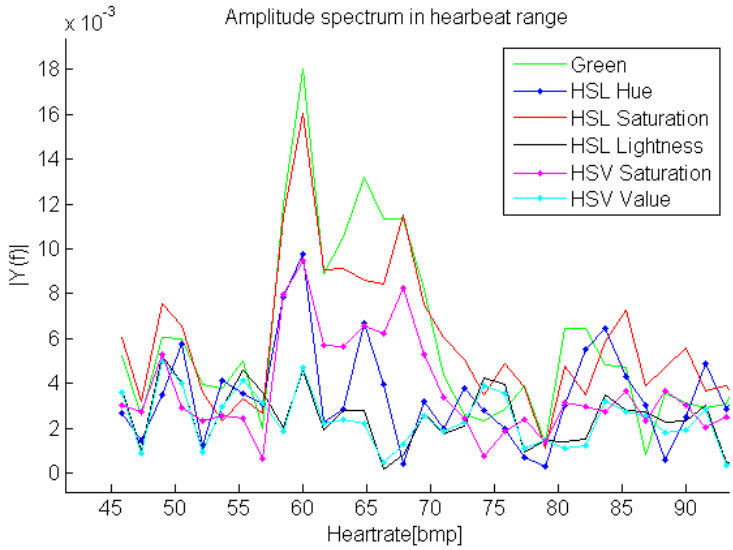
In most previous non-contact PPG measurements and experiments tests have been performed on subjects with light, or Caucasian skin tones. This experiment was designed to test if darker skin tones had any effect on the measurement of HR using non-contact PPG. Figure 5.9 shows experiment setup and ROI placement. One video was recorded and from this video three individual samples of 1000 frames was extracted. Observe figure 5.10 for frequency spectra taken from the three samples. During the experiment HR was detected using a pulse oximeter. For sample 1 (a) the oximeter measured HR at 59 bpm., for sample 2 (b) the oximeter measured HR at 61 bpm. and for sample 3 (c) the oximeter measured HR at 62 bpm. For figure 5.10(a) and (b) the frequency ranges are in the entire HR range (40-200 bpm.) whereas in (c) the frequency range is limited to were the resting pulse is expected (40-100 bpm.) for a closer examination of this region. In (a) we have removed the HSV color channels for clarity.



(a) FFT of run 1. Oximeter measured HR at 59 bpm. HSV removed for clarity.



(b) FFT of sample 2. Oximeter measured 61 bpm.



(c) FFT of sample 2. Oximeter measured 62 bpm. Showing a limit frequency range for closer examination.

Figure 5.10 – FFT of three runs, 1000 consecutive frames each, on non-caucasian subject. Different colored plots are for the different color space channels used for measuring HR.

5.6 Phase difference in HR measurements

In this section the results regarding measurements of phase difference in HR in the case of vascular occlusion[16] will be present. It is theorized that if a extremity of the human body has a vascular occlusion, a phase delay relative to an unoccluded pulse wave can be observed. This is because an occlusion should cause a drop in blood pressure distal from the occlusion point and this will cause an decrease in pulse wave velocity (PWV) and increase in pulse transit time (PTT) distal from the occlusion point.

To measure a phase difference between two waves, an unoccluded point to measure relative to is needed. In the case of no occlusion, a PPG wave at points at a similar distance from the heart, on the left and right side of the body, should be in phase. The human body is not totally left-right symmetrical, but any discrepancies caused by this should be negligible.

Experiment 1 (5.6.1) was designed to test if two points at a similar distance from the heart, in this case the palm of the hand on the left and right side of the body, had PPG signals in phase with each other.

Experiment 2 (5.6.2) was designed to test what a full occlusion of one of the extremities would do with the phase relative to an identical point on the other side of the body. It was also designed to test what happened with the signal in the occluded arm when the occlusion was suddenly removed.

Experiment 3 (5.6.3) was designed to test what happened to the phase difference when the occlusion in the arm was gradually increased.

The last experiment presented in this section (5.6.4) was an experiment to test if it was possible to detect phase difference between feet. In this experiment the right foot was occluded while the underside of the feet were filmed.

HR during the experiment was measured using pulse oximeter and ECG.

5.6.1 Experiment 1

The first test was to observe the HR phase difference is between two points at a similar distance from the heart, but on opposite sides of the body.

In this test the subject was sitting in a chair with a resting pulse. The quality of the video was less ideal case. Two ROIs were placed on the right and left palm of the hand. The test setup is shown in figure 5.11, placement of ROIs are shown with a red frames. The test video was divided into two parts, each with 1000 consecutive frames at 25 fps. (40 second recordings). Frequency spectra of the first recording are shown in figure 5.12. 5.12(a) shows the entire frequency spectrum while (b) shows frequency spectra confined to the resting pulse spectrum for a closer inspection. A trace of the Green RGB channel can be observed in figure 5.13. The trace was bandpass filtered in the resting HR range (50-100 Hz) to remove noise and over harmonic components and to visually observe the phase difference. The HR was measured at 77 bpm. during both experiments. The next 1000 frames displayed similar results and are presented in appendix A.1.

[16] Vascular occlusion means that the blood vessels are blocked, i.e. no or little blood can flow through the blockage

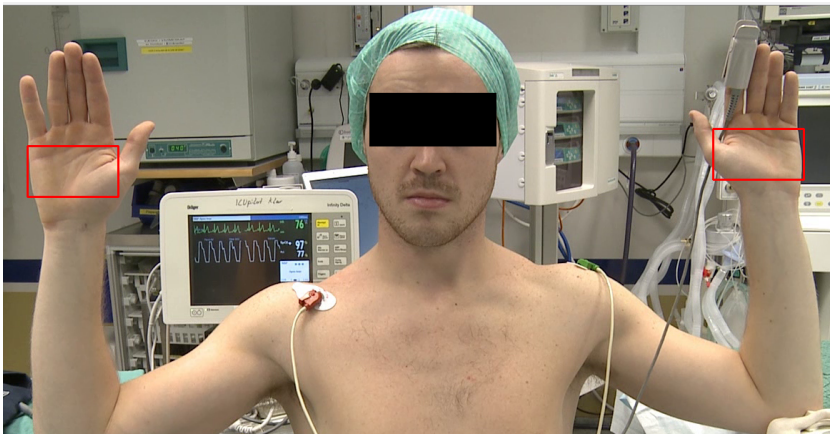
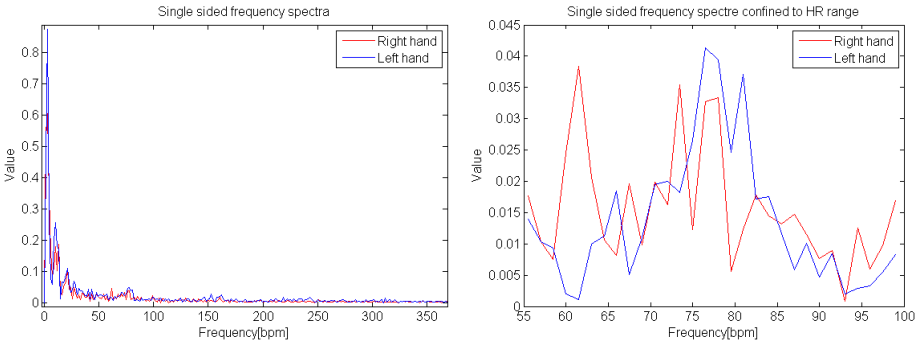


Figure 5.11 – Test setup for experiment 1. ROI is red rectangles on each hand.



(a) Frequency spectra

(b) Frequency spectra confined to HR range

Figure 5.12 – Frequency spectra of experiment 1, benchmark test. HR measured to 77 bpm. by pulse oximeter.

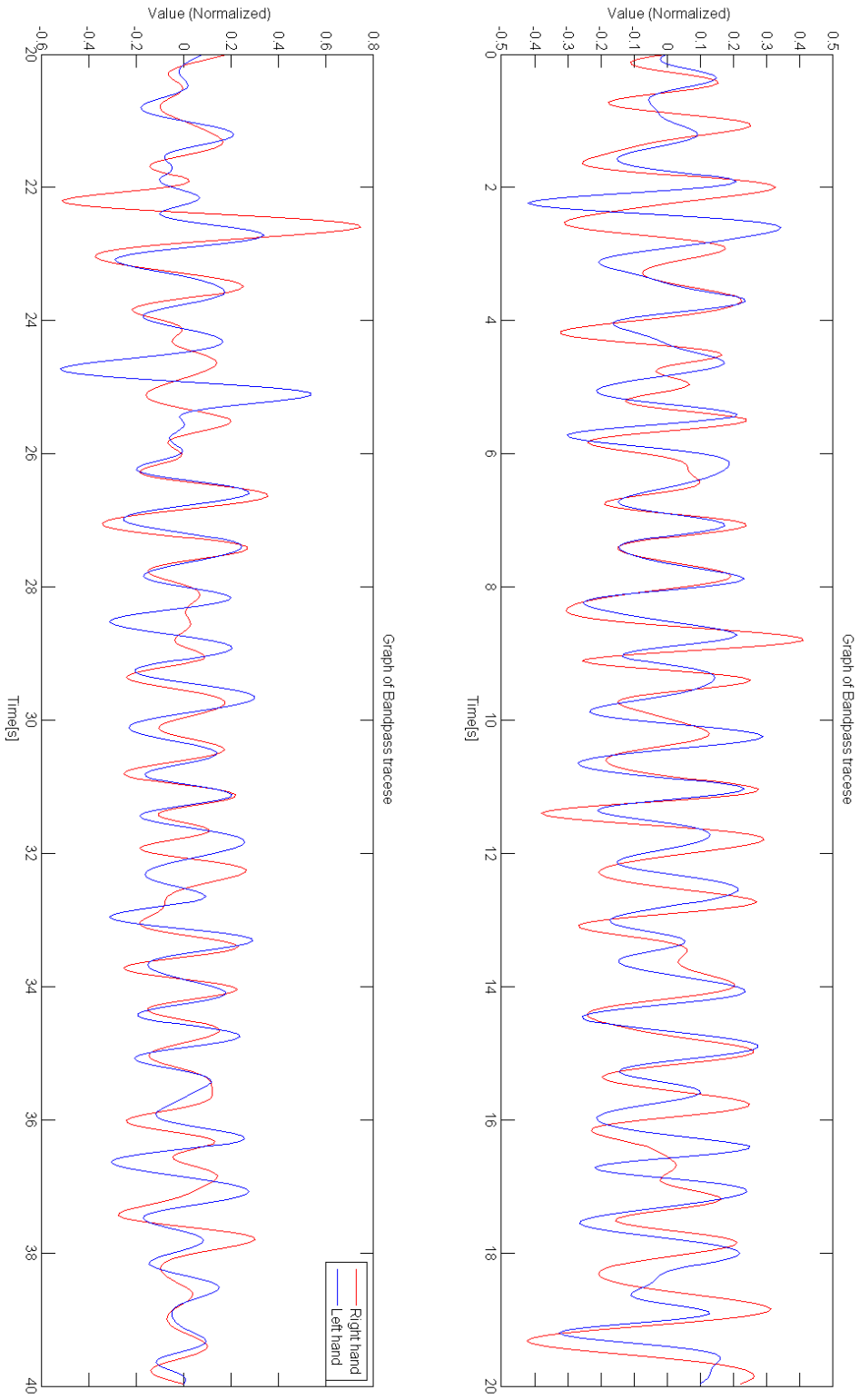


Figure 5.13 – Experiment 1, benchmark test. Trace of change in Green channel, bandpass filtered in HR range

5.6.2 Experiment 2

Experiment 2 had a similar setup to experiment 1. The main difference was that the subject has a medical cuff strapped tightly around the right arm, just below the armpit, as can be observed in figure 5.14. ROIs were placed over each hand and are shown as red rectangles in the figure. The medical cuff caused a strong occlusion of the right arm while the left hand was unoccluded. The cuff was on during the first half of the video, then removed halfway through. The results were divided into three part so to be viewed individually.

In part 1 of 3 the cuff is present and there is a strong occlusion in the right arm. A bandpass filtered signal can be observed in figure 5.18 and frequency spectra can be observed in figure 5.15. The first spectrum shows the entire frequency range while the second shows the HR range for a closer inspection of the more important frequency ranges.

In part 2 of 3 the occlusion is suddenly released after 14 seconds. Trace can be observed in figure 5.19. Frequency spectra in figure 5.16.

In part 3 of 3, one can observe the period after the occlusion is removed. Trace can be observed in figure 5.19 and frequency spectra in figure 5.17.

The HR was measured at 77 for the first two parts, and 75 for the last part. This experiment was repeated and the results of this can be observed in the appendix chapter A.2.

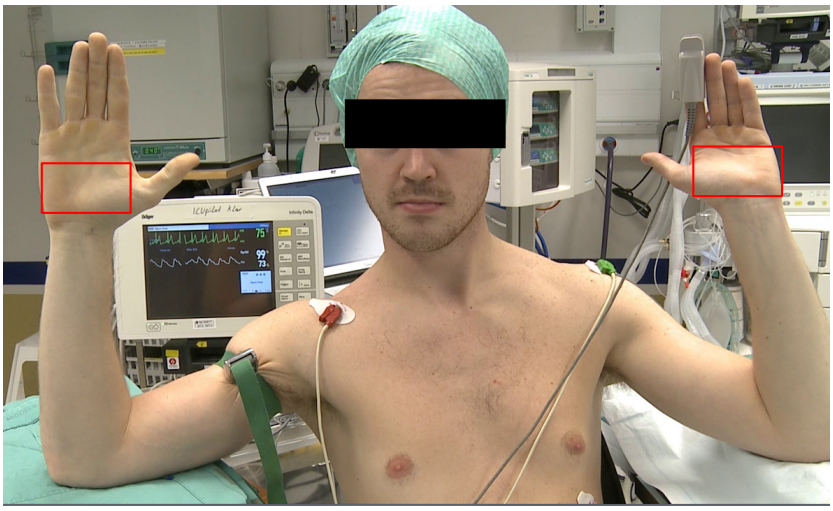
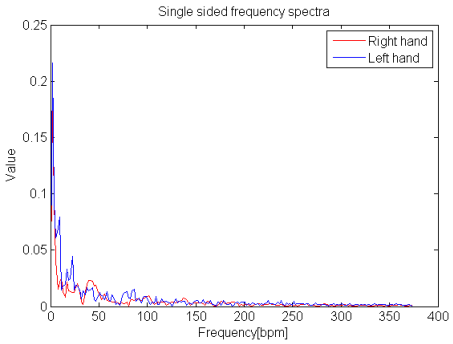
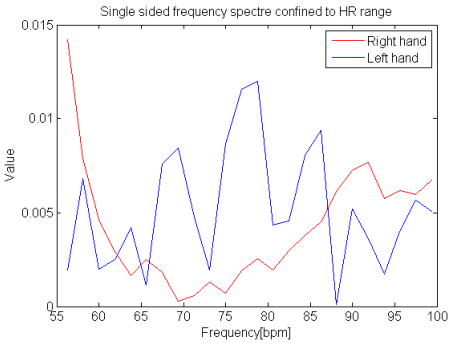


Figure 5.14 – Test setup for experiment 2. ROI are red rectangles on each hand. Notice medical cuff on right arm for occlusion

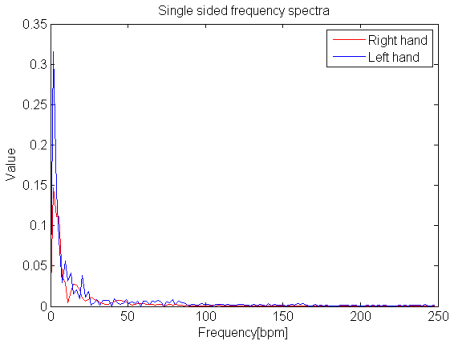


(a) Frequency spectrum



(b) Frequency spectrum confined to resting HR range

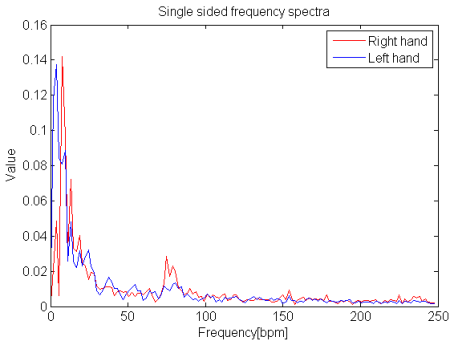
Figure 5.15 – Frequency spectra of experiment 2, part 1 of 3. The Right hand is fully occluded.



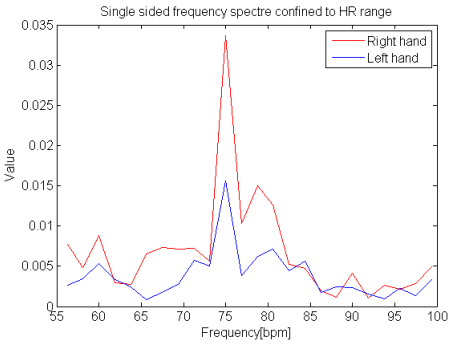
(a) Frequency spectrum

(b) Frequency spectrum confined to resting HR range

Figure 5.16 – Frequency spectra of experiment 2, part 2 of 3. Occlusion is released 14 seconds into the recording.



(a) Frequency spectrum



(b) Frequency spectrum confined to resting HR range

Figure 5.17 – Frequency spectra of experiment 2, part 3 of 3. Occlusion is removed.

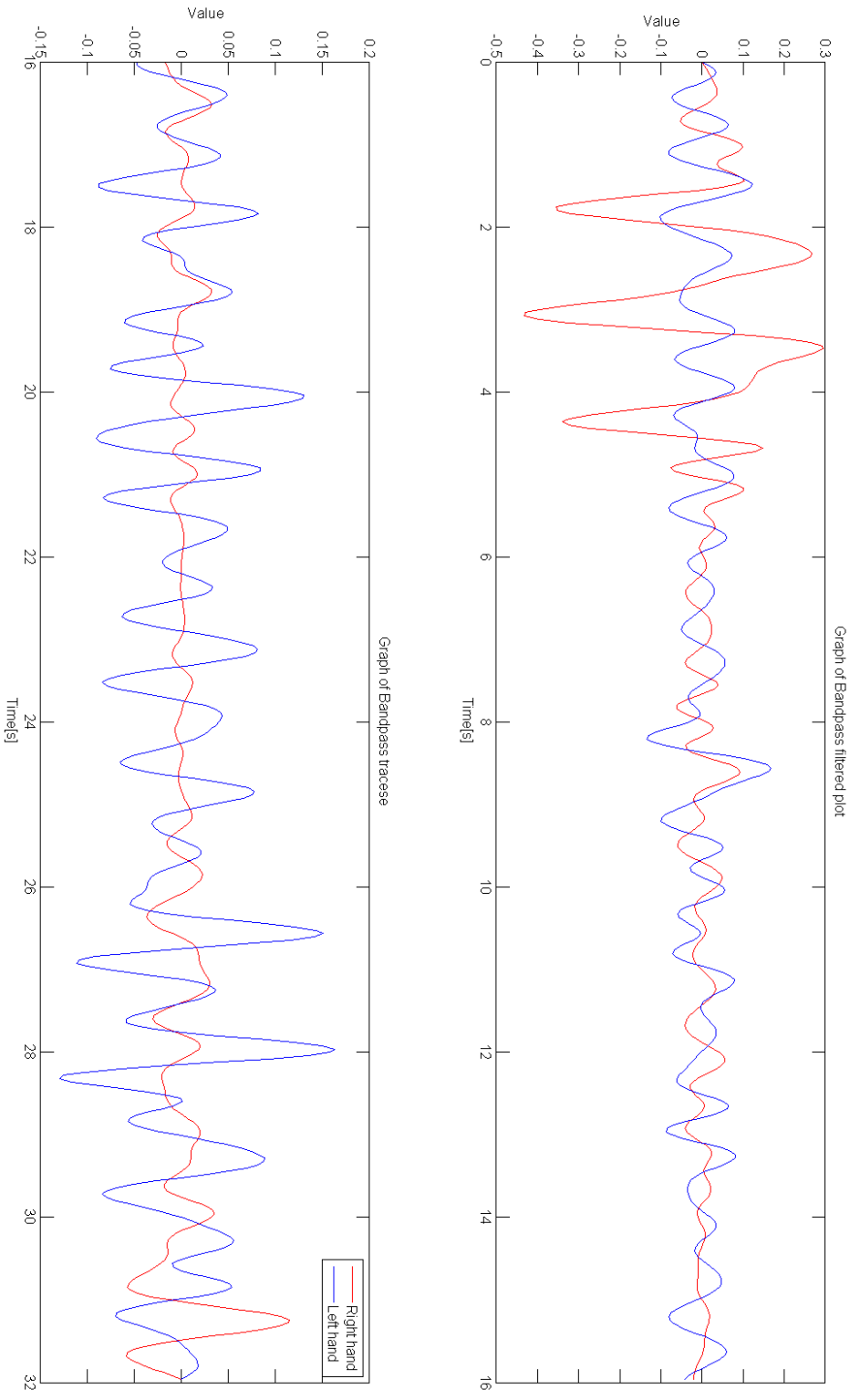


Figure 5.18 – Experiment 2, part 1 of 3. Right hand fully occluded. Trace of change in Green channel, bandpass filtered in HR range

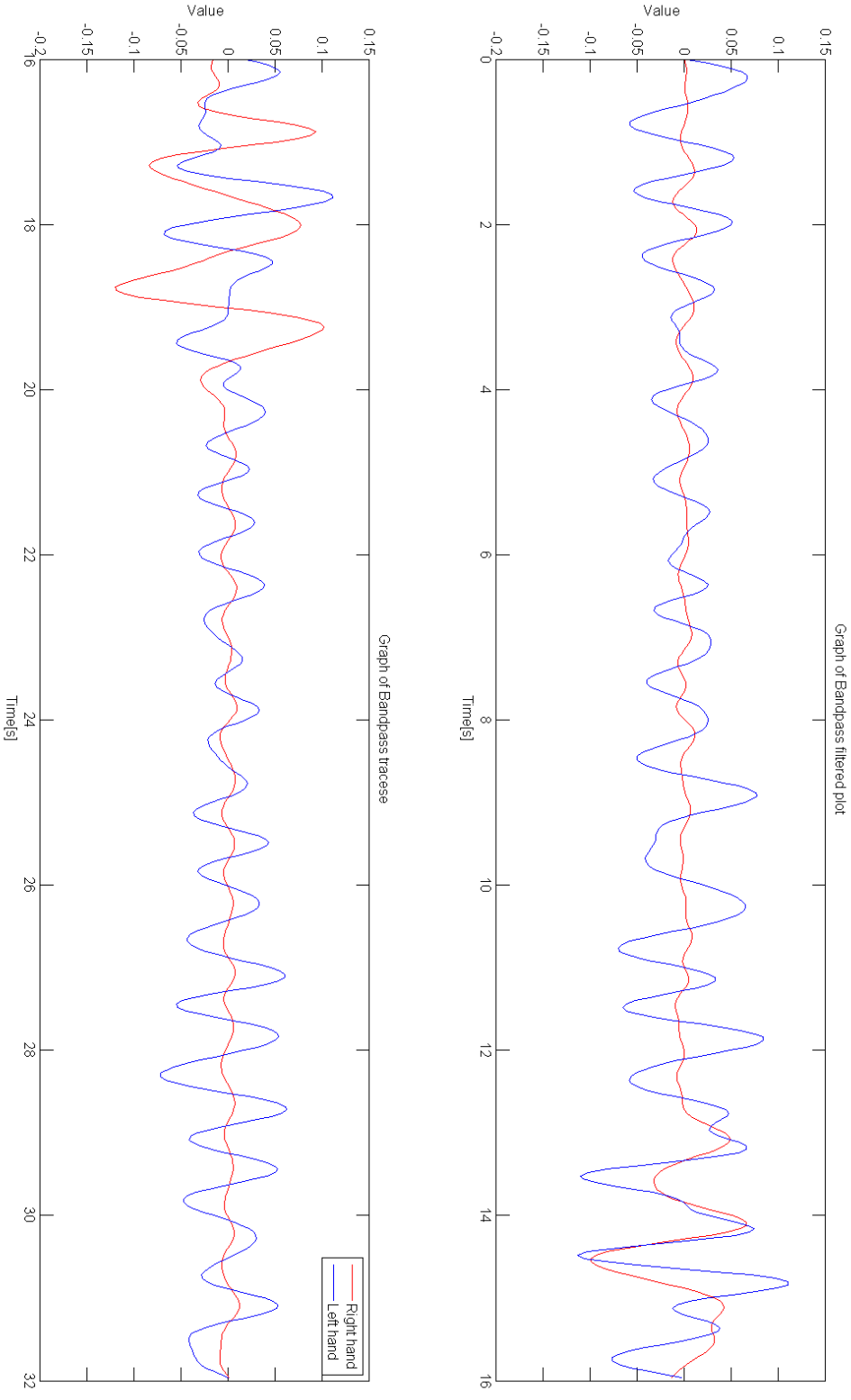


Figure 5.19 – Experiment 2, part 2 of 3. Right hand is fully occluded cuff is removed at the 14 s mark. Trace of change in Green channel, bandpass filtered in HR range

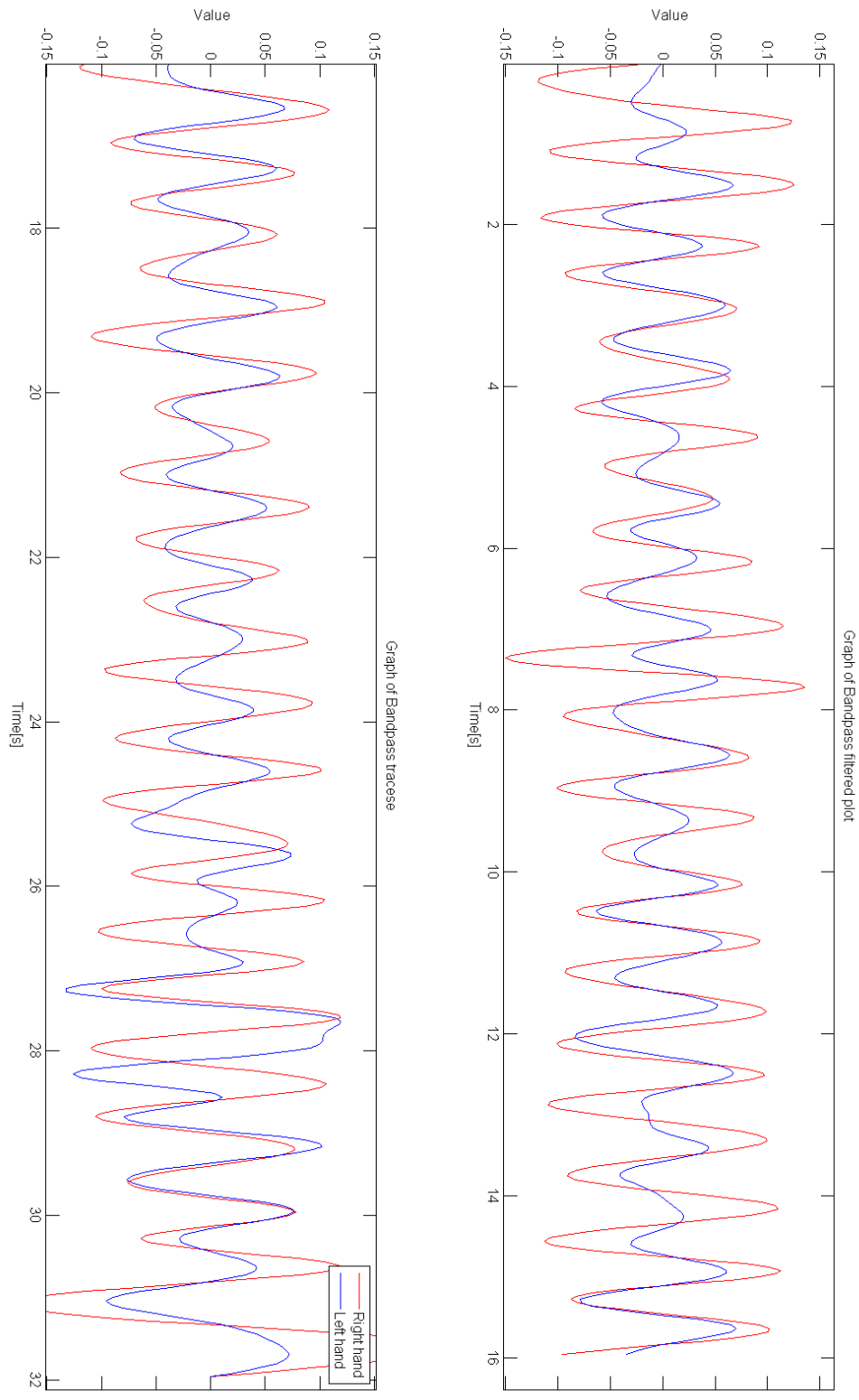


Figure 5.20 – Experiment 2, part 3 of 3. Medical cuff removed and there is no occlusion. Trace of change in Green channel, bandpass filtered in HR range

5.6.3 Experiment 3

Experiment 3 had a similar setup to experiment 1 and 2. In this case a medical cuff which could be gradually inflated was used. This was done to investigate if it was possible to observe changes in the HR pulse phase as a function of the degree of occlusion. Figure of setup can be observed in figure 5.21.

The medical cuff was inflated over a period of approx. 30 seconds. The results are divided into three parts of 40 seconds each, with some overlap between part 2 and 3. In part 1 of 3 the cuff is not inflated in order to establish a baseline for comparison. In part 2 of 3 the cuff starts to gradually inflate until a maximum level is reached after 20 seconds. After approximately 30 seconds the cuff deflates and the occlusion is removed. Part 3 has some overlap with part 2 and one can observe the deflation of the cuff in the start of part 3.

In part 1 oximeter and ECG measured HR between 70 and 75 bpm. In part 2 and 3 the HR was measured at 75 bpm.

Experiment 3 was repeated four times. In two of these attempts noise obscured all results. The results from the last attempt can be observed in appendix chapter A.3.

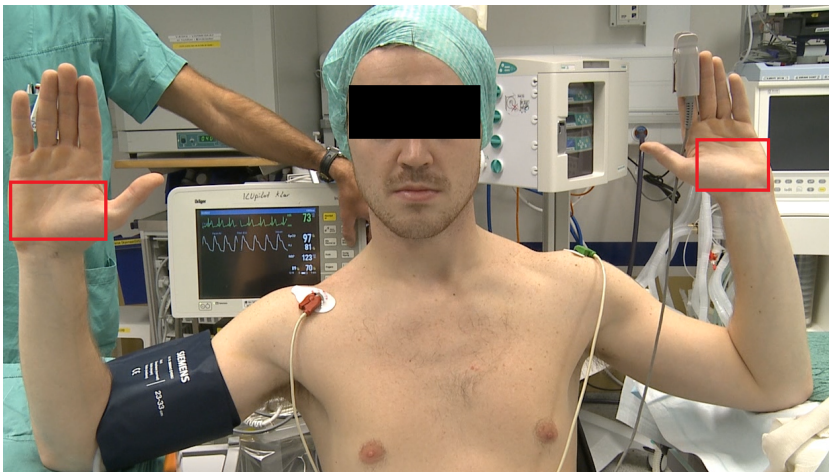


Figure 5.21 – Setup experiment 3. ROI are red rectangles on each arm. Note medical cuff of right arm which could be gradually inflated.

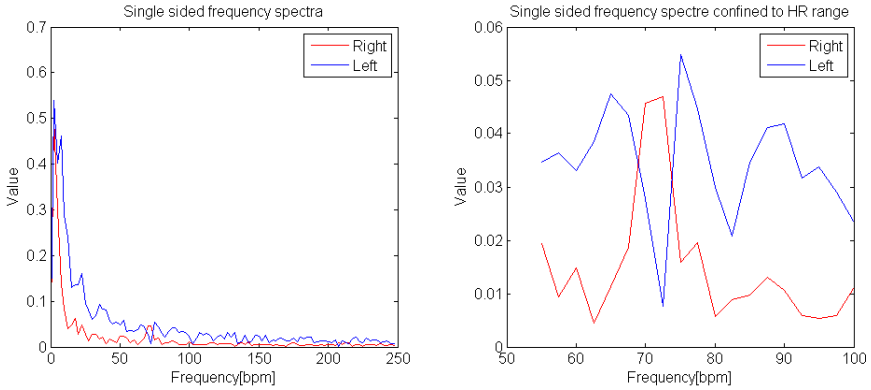


Figure 5.22 – Frequency spectra of experiment 3, part 1 of 3. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

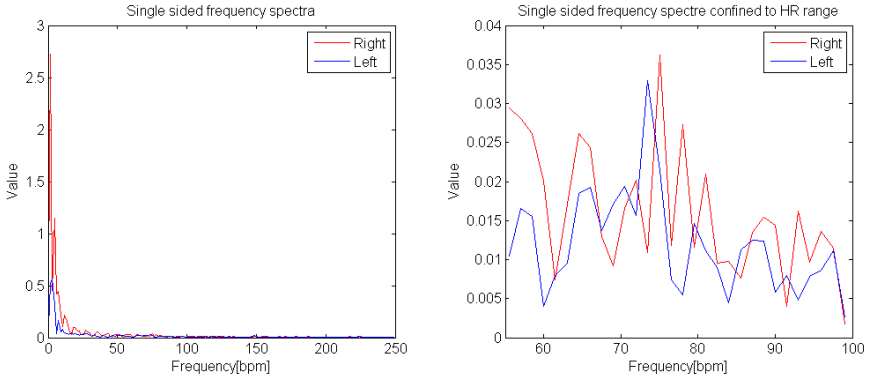


Figure 5.23 – Frequency spectra of experiment 3, part 2 of 3. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

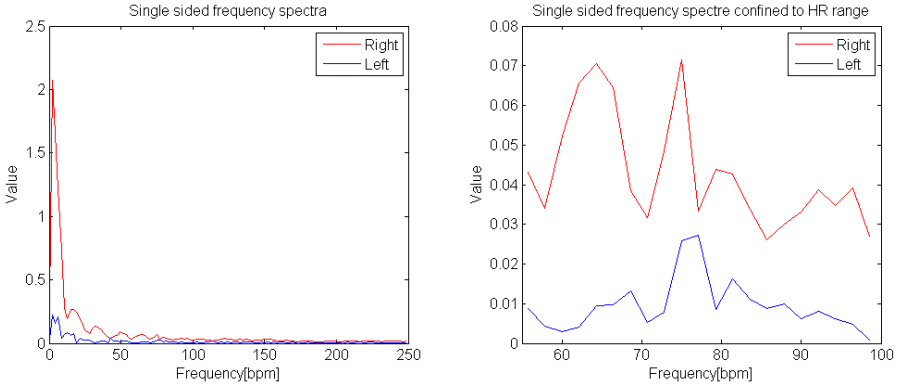


Figure 5.24 – Frequency spectra of experiment 3, part 3 of 3. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

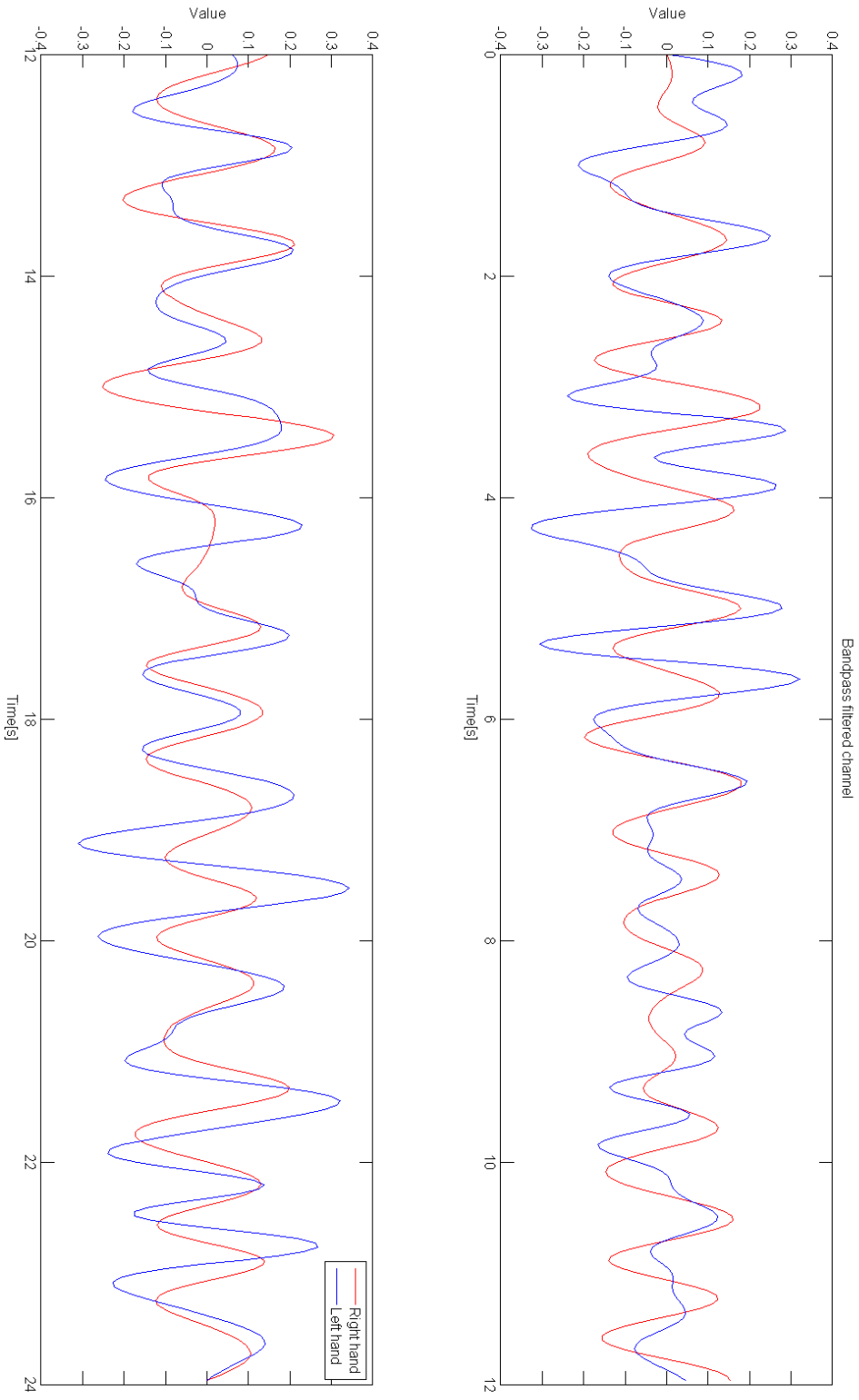


Figure 5.25 – Experiment 3, part 1 of 3. Cuff is not inflated and there is no occlusion. Trace of change in Green channel, bandpass filtered in HR range

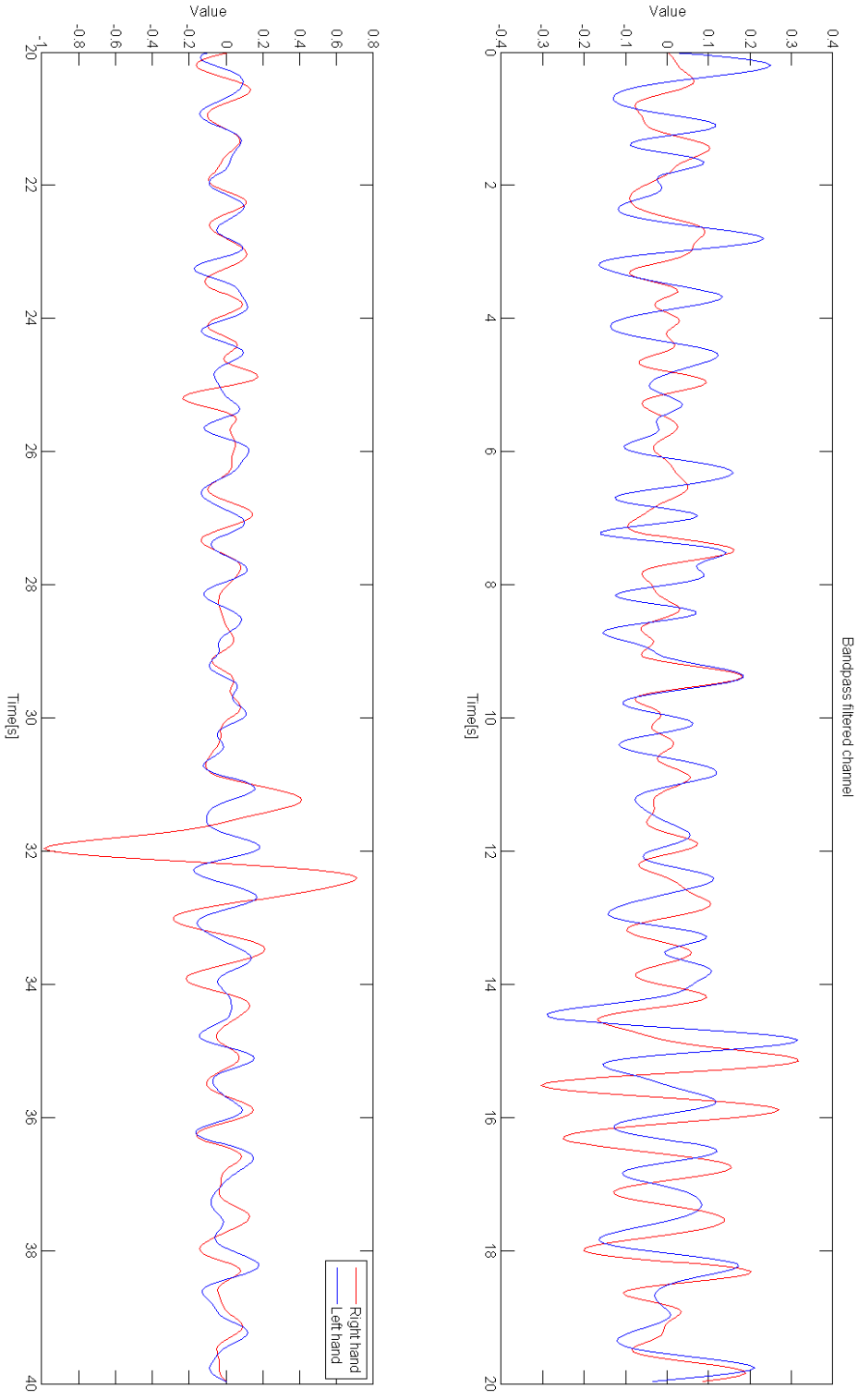


Figure 5.26 – Experiment 3, part 2 of 3. The medical cuff is gradually inflated over a period of 20 seconds and is deflated after 30. Trace of change in Green channel, bandpass filtered in HR range

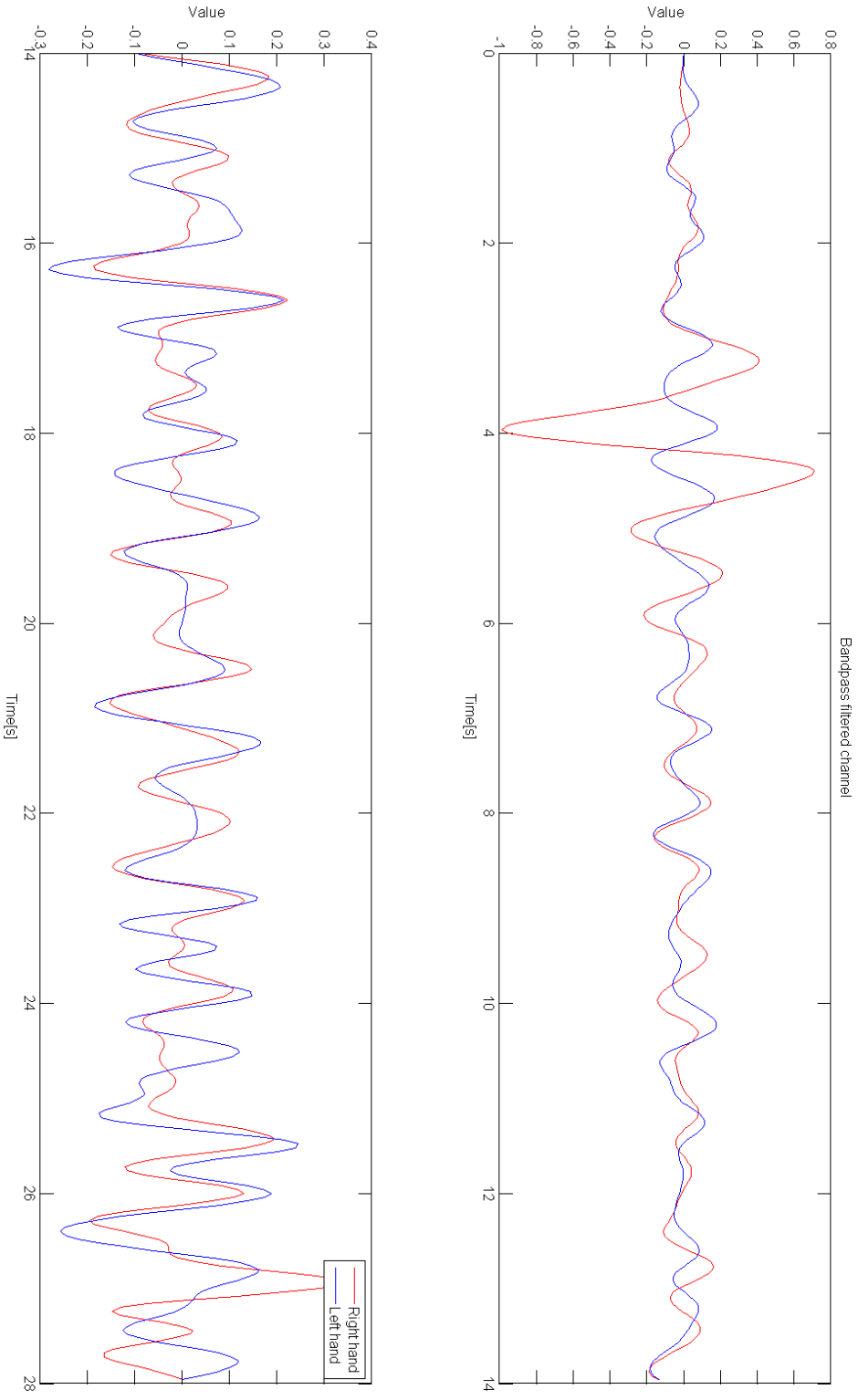


Figure 5.27 – Experiment 3, part 3 of 3. Cuff is deflated at the 2 second mark and the right arm is unoccluded. Trace of change in Green channel, bandpass filtered in HR range

5.6.4 Phase difference in feet

Since satisfying results were obtained for HR detection under the feet, as shown in section 5.4, experiments to detect phase shift with occlusion in one of the feet were performed.

This experiment had the same setup as shown in figure 5.5. In this case a medical cuff was used to induce an occlusion in the right foot. The experiment results were divided into three parts: In the two first the right foot is occluded while in the last part the occlusion was removed. HR was measured using pulse oximeter. The pulse was somewhat varying, but usually found at around 69 bpm. for all parts of the experiment.

In figure 5.31 and 5.29 a trace of the Green channel while the right foot is occluded can be observed while figure 5.33 shows the unoccluded trace. All traces are bandpass filtered in HR range. Figures 5.28, 5.29 and 5.30 shows the frequency spectra of the three parts respectively. The left of the figures shows the entire frequency spectra, while the right is confined to resting HR range for closer inspection.

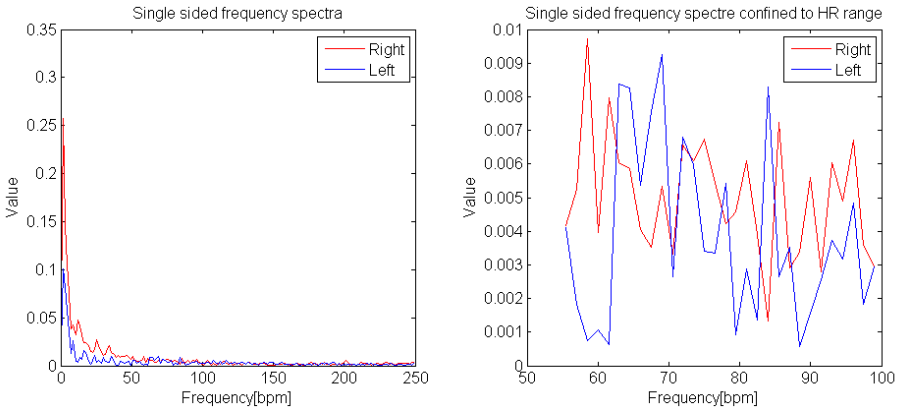


Figure 5.28 – Frequency spectra of underside of feet phase difference experiment, part 1 of 3. Right foot occluded. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

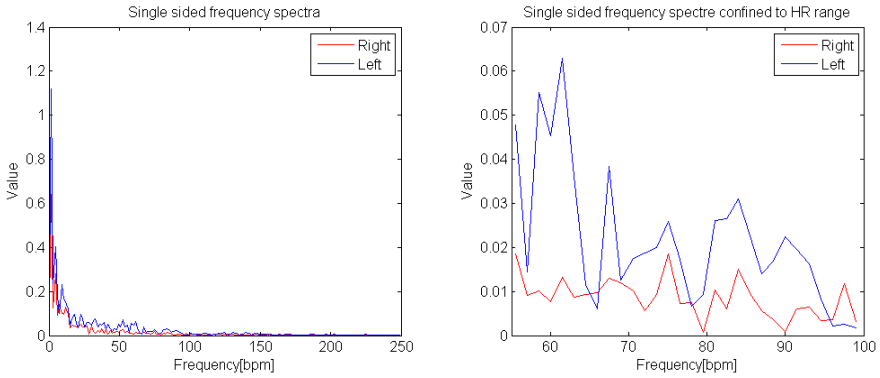


Figure 5.29 – Frequency spectra of underside of feet phase difference experiment, part 2 of 3. Right foot occluded. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

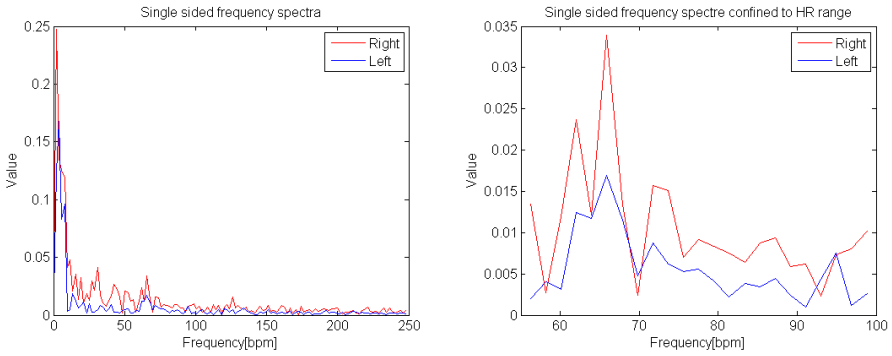


Figure 5.30 – Frequency spectra of underside of feet phase difference experiment, part 3 of 3. No occlusion. The left figure shows the entire frequency range while the right is confined to resting HR range for closer observation.

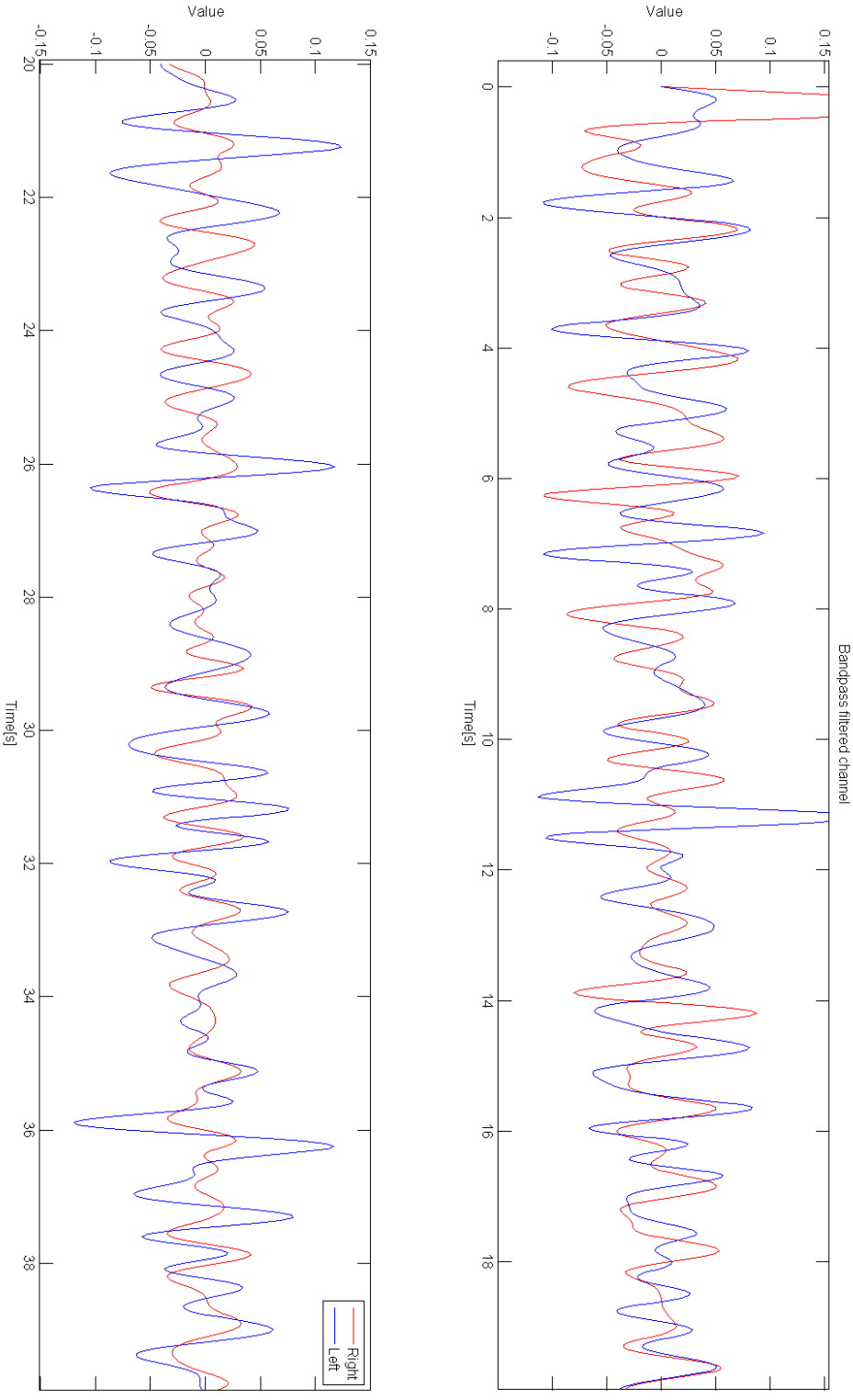


Figure 5.31 – Trace from underside of feet phase difference experiment, part 1 of 3. Green channel of RGB color space, trace is bandpass filtered in HR range. Right foot occluded.

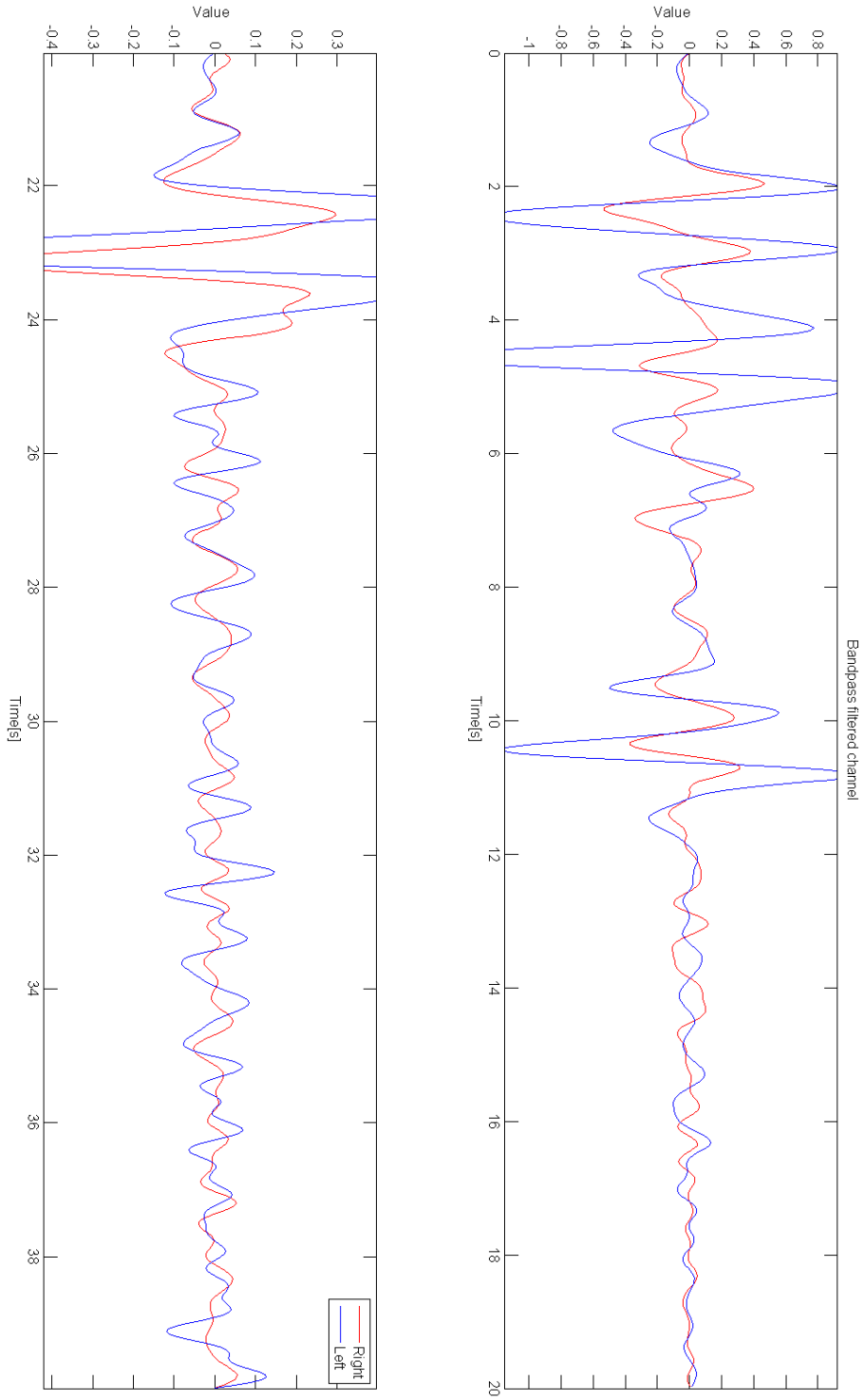


Figure 5.32 – Trace from underside of feet phase difference experiment, part 2 of 3. Green channel of RGB color space, trace is bandpass filtered in HR range. Right foot occluded.

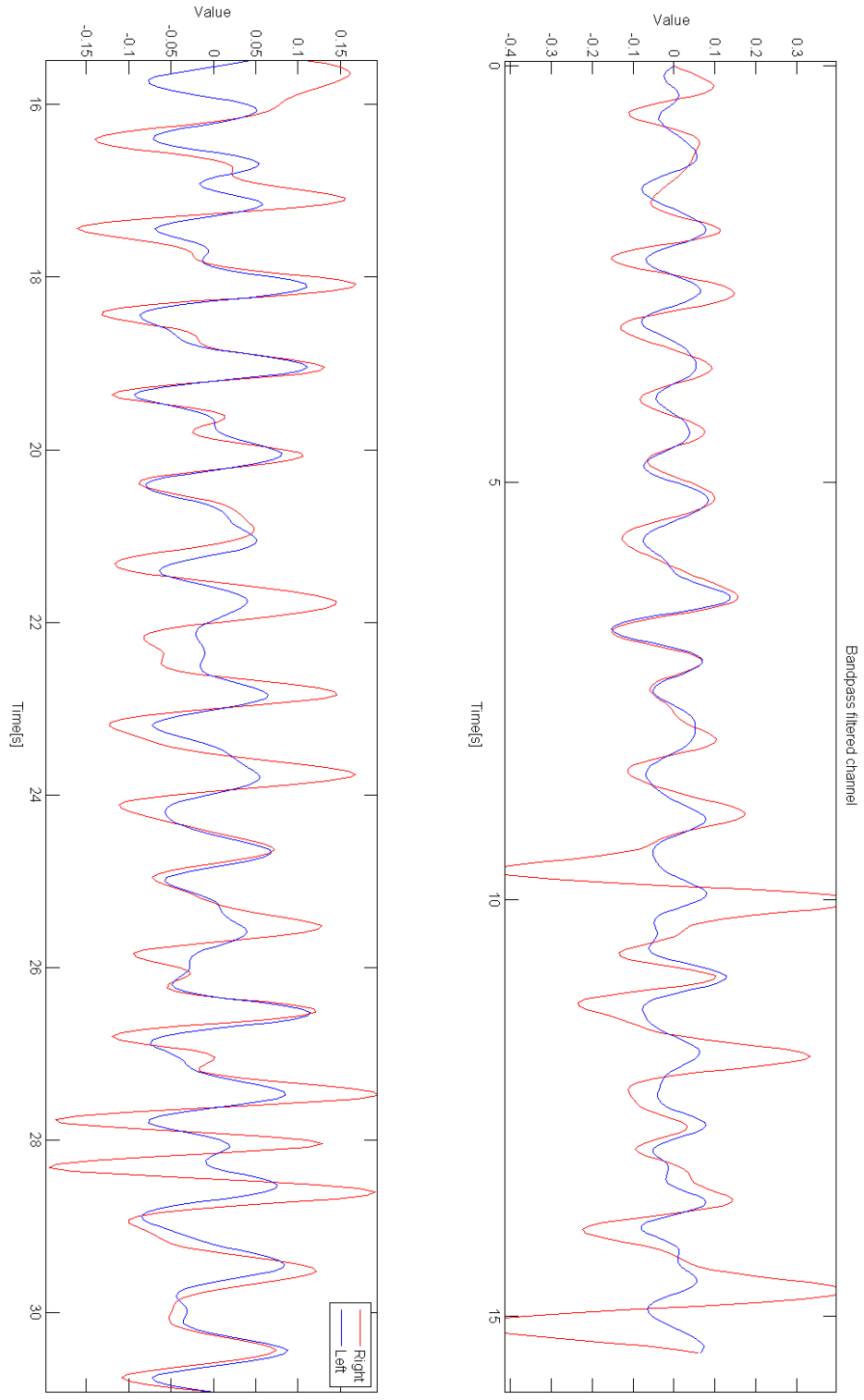


Figure 5.33 – Trace from underside of feet phase difference experiment, part 3 of 3. Green channel of RGB color space, trace is bandpass filtered in HR range. No occlusion.

5.7 Synchronization of PPG with ECG data

In this section we investigate how the PPG signal is correlated to the ECG of a subject. This was done done by filming a subject while simultaneously using a digital ECG to record ECG data. An ROI was placed over the subjects forehead and a trace of the Green channel from the RGB color space was taken. This trace was bandpass filtered using a zero phase filter in HR range and synchronized with the ECG data. The video was recorded in a similar manner to that which can be observed in figure 5.13, only in this case the ROI was placed over the subjects forehead.

Synchronization was achieved by two methods. First by physical input that shows up on both signal sources[17]. Secondly both timestamps on the ECG recording and the video were synchronized at the start of recording.

These experiments were performed under three different circumstances for the subject.

In the first case there was no occlusion present (5.7.1).

In the second case a gradually increasing occlusion was induced on the subjects arm (5.7.2)

In the last case a full occlusion and subsequent release was induced on the subject (5.7.3).

The experiments were performed under different circumstances because we wanted to observable if the ECG changed because of these circumstances, and if so, how this change in ECG affected the PPG signal.

5.7.1 No occlusion

In this experiment there was no occlusion present. The video recording were synchronized with the ECG data and the results can be observed in figure 5.34. Both signals are normalized to a -1 to 1 value range for easier comparison. A strike against one of the ECG probes was used as a synchronization signal and can be observed as a disturbance from 0 to 4 seconds into the recording in both signals. The bandpass filter used was a zero phase filter so that we do not get a phase delay relative to the ECG signal.

5.7.2 Gradual increasing occlusion

In this experiment we wanted to investigate if there is a detectible change between ECG and the PPG signal when we induce an gradual occlusion to one of the arms.

The experiment setup was identical to those in the no occlusion experiment shown in section 5.7.1, except that the occlusion was gradually increased on the right hand over a time period of 40 seconds followed by a subsequent release. The results from the experiment can be observed in figure 5.35. The PPG signal is from the Green channel of the RGB color space and is bandpass filtered in the HR range using a zero phase filter. Both signals are normalized to -1 and 1 value range.

[17] "Physical input" i.e. whacking one of the ECG probes. Whacking was observable both on video and in the ECG recording.

5.7.3 Sudden removal of occlusion

The experiment was identical to the gradual increasing occlusion experiment shown in subsection 5.7.2, except here the occlusion on the right hand of the subject is full at the start of the video and then removed approx. halfway through. The results can be observed in figure 5.36. The occlusion is removed approx. at the 47 second mark. The PPG signal is from the Green channel of the RGB color space and is bandpass filtered in the HR range using a zero phase filter. Both signals are normalized to -1 and 1 value range.

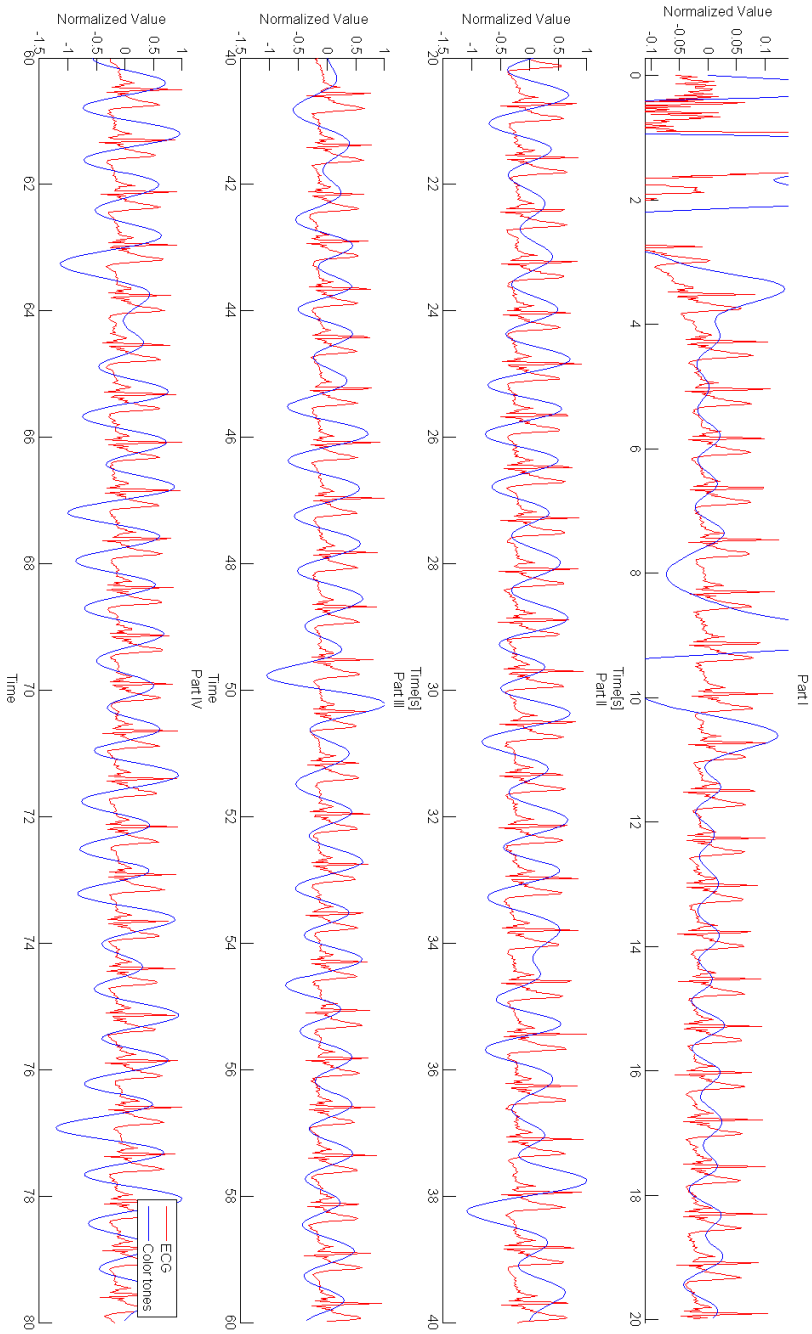


Figure 5.34 – Trace of synchronization without occlusion experiment. ECG and trace of Green RGB channel synchronized. Trace is bandpass filtered in the HR range using zero phase filter. No occlusion present. Disturbance viewed from 0-4 seconds in both signals are due to input used to synchronize the signals.

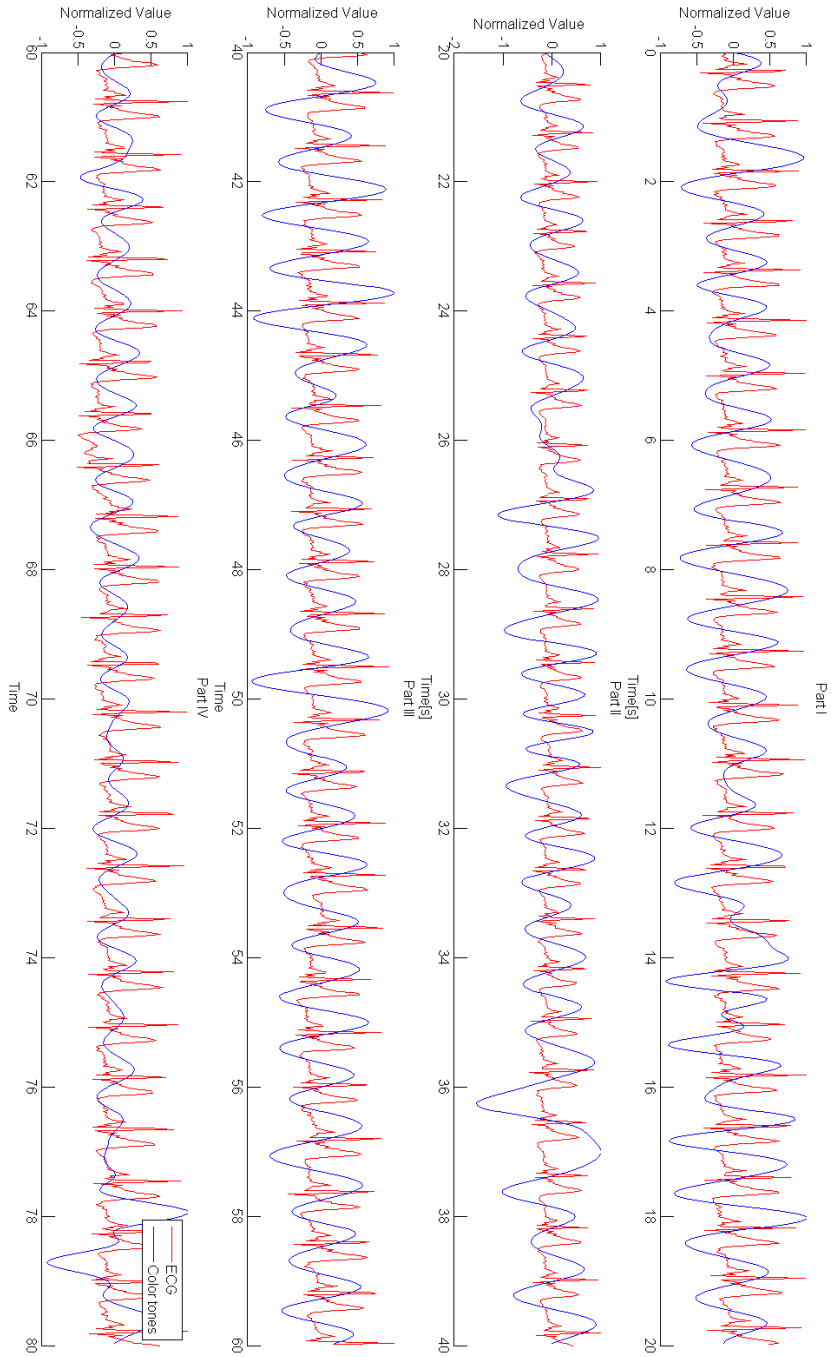


Figure 5.35 – Trace of synchronization with gradually increasing occlusion experiment. ECG and trace of Green RGB channel synchronized. Trace is bandpass filtered in the HR range using zero phase filter. Occlusion is gradually increased from the 20 second mark and reaches full occlusion at the 40 second mark. Occlusion is released at the 60 second mark.

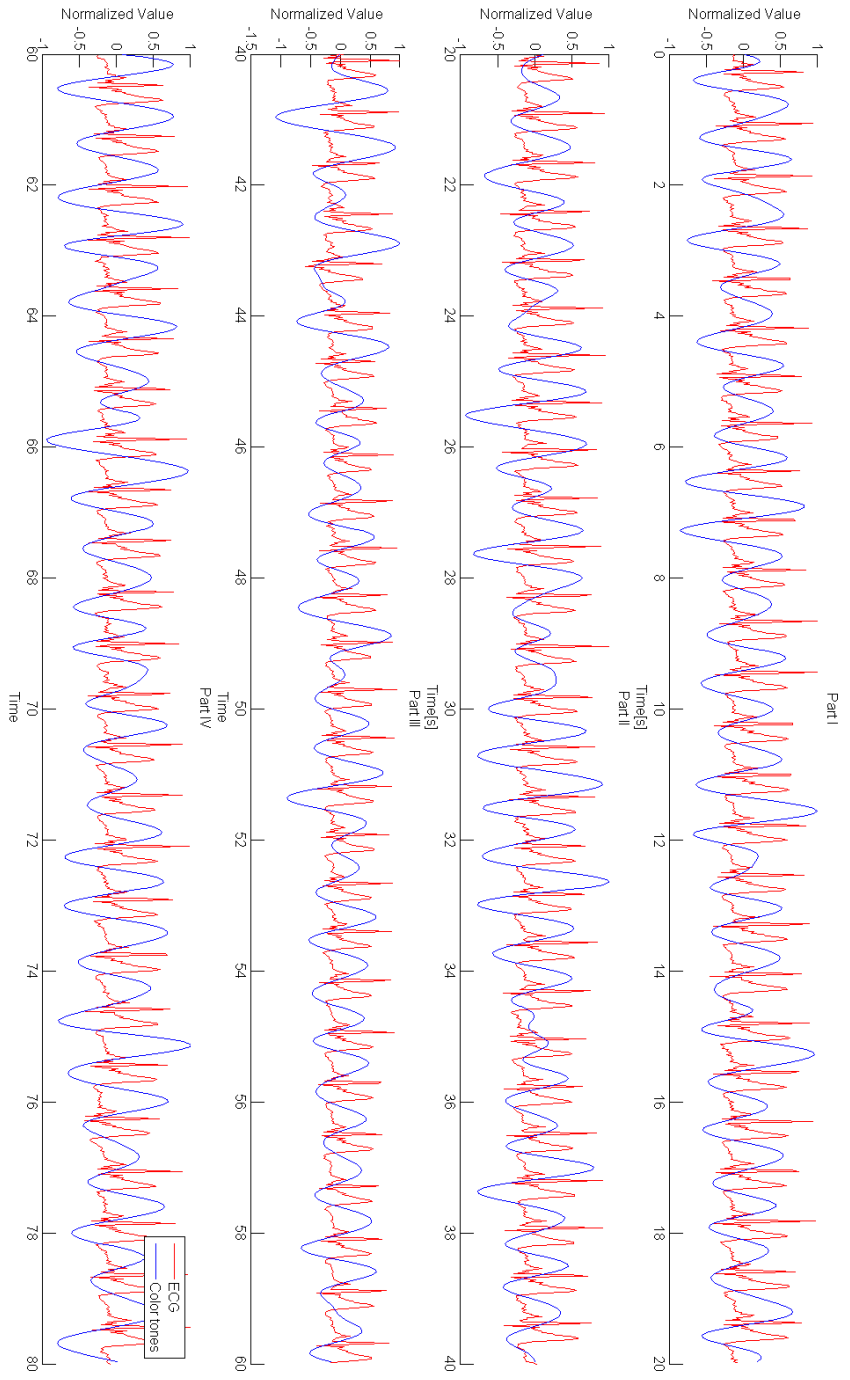


Figure 5.36 – Trace of synchronization with sudden removal of occlusion experiment. ECG and trace of Green RGB channel synchronized. Trace is bandpass filtered in the HR range using zero phase filter. The video start with a full occlusion of the right arm. The occlusion is then removed at the 47 second mark.

5.8 Phase difference in laparoscopy and colonoscopy video.

It is theorized that cancerous growths in tissue may exhibit different blood flow characteristics than the tissue surrounding it. In this case it may present itself as a phase difference in the PPG signal relative to PPG signals of the surrounding tissue.

It is also theorized that PPG signals could be used to detect whether blood circulation into a transplanted organ had started successfully after an organ transplant. When a patient receives a transplanted organ, it is not always certain that blood circulation into the donated organ resumes correctly. If blood circulation into a newly operated organ does not start correctly after surgery, necrosis[18] in parts of the organ can occur. If a PPG signal from video of an internal organ could automatic detect this, it could be a useful tool for medical professionals after organ transplant surgery.

To test these two theories, videos from laparoscopy and colonoscopy were analyzed. The colonoscopy videos showed a cancerous growth in the liver, whereas the laparoscopy videos showed different transplanted lungs after it had been operated into patients.

Unfortunately the laparoscopy videos proved too unstable to obtain a satisfying video recording to extract data from. This due to the constant inhale and exhale of the lungs, disturbing the video such that getting a stable recording for the required length proved itself too difficult.

The laparoscopy videos were more stable, and a reasonably stable 7 second clip showing a cancerous growth during a liver laparoscopy was extracted. Tracking algorithms were used to get a relatively stable recording of ROIs, of which one contained a cancerous growth. The Hue channel from HSL color space was found to give the best results. A frame from the laparoscopy videos used is shown in figure 5.37. The figure also shows the ROIs used. In figure 5.38 one can observe the frequency spectra of the different ROIs and in figure 5.39 one can observe a bandpass filtered trace of the Hue channel of each of the ROIs.

[18] Premature cell death in living tissue

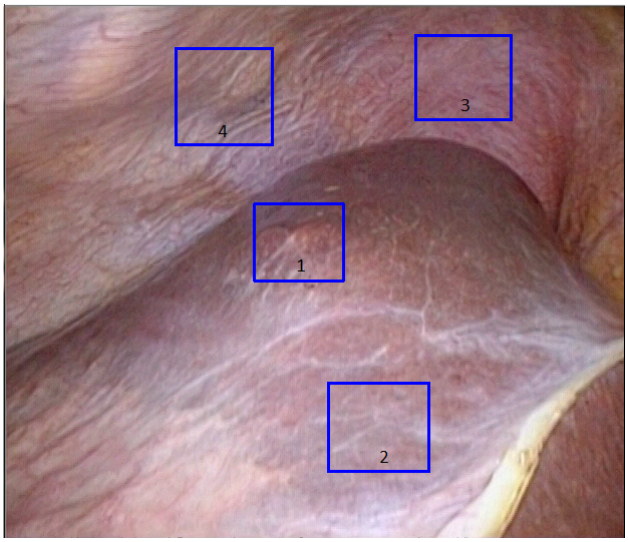


Figure 5.37 – Video frame of the liver laparoscopy. ROIs used are shown with a blue frame. ROI 1 contains the cancerous growth while ROI 2-4 were used for comparison.

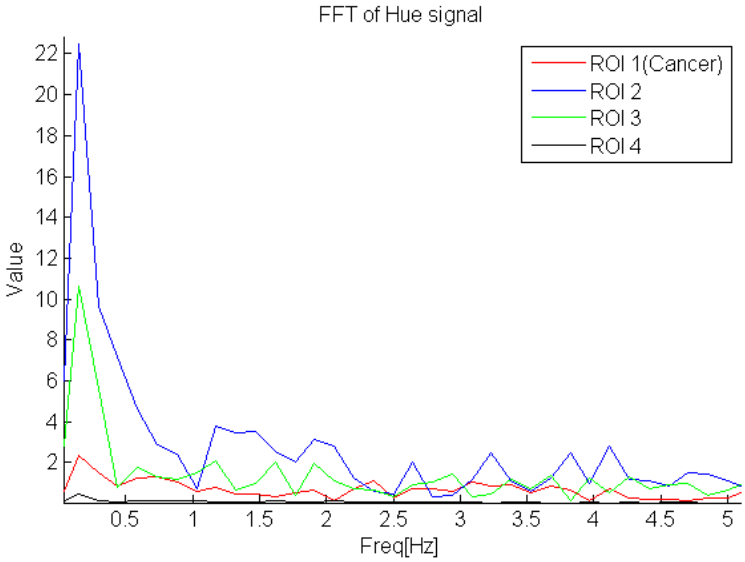


Figure 5.38 – Frequency spectra of the different ROIs. Taken over 170 frames.

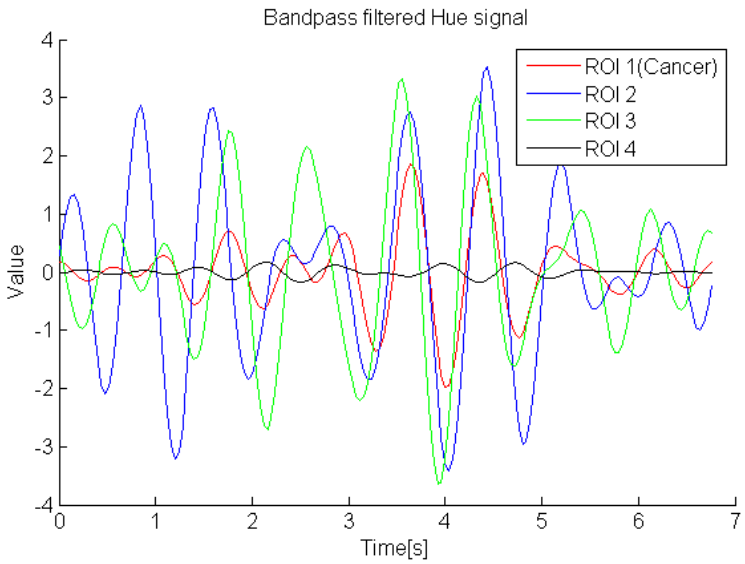


Figure 5.39 – Trace of the Hue channel in the different ROIs. Taken over 170 frames. Hue channel bandpass filtered in HR range.

6 Discussion

In this section we will discuss and review the results presented in the result section and the work on the functional non-contact PPG system.

The discussion will happen in mainly the same order as the results are presented in the results chapter. The main difference is that the results presented in the comparison of different color space section (5.3) will be discussed in the last section in this chapter along with a general discussion of the non-contact PPG system.

In subsection 6.1 we will discuss the effects and efficiency of ICA on non-contact PPG systems.

In subsection 6.2 we will discuss the results from the medical experiments done at Rikshospitalet and presented in result chapters 5.4 to 5.7.

In subsection 6.3 we will discuss the C++ framework developed and the results it gave. We will analyze the different color spaces against each other. We will also discuss the further development of a functional non-contact PPG system to detect HR.

If a special property or characteristic of a PPG signal have reviled themselves in the course of an experiment, we will in some cases discuss them although they may not be directly relevant to the experiment in that section in itself.

In this chapter, when we refer to something as either a success or a failure, we mean that as short for that a process or a channel succeeded or failed at detecting a HR signal and present the correct HR frequency relative to a pulse oximeter measurement.

6.1 The use of ICA on PPG

In this section we will discuss the use of ICA in PPG systems, and its efficiency in different color spaces.

6.1.1 Cross correlation between RGB and HSL channels

In the experiment section we first investigated the cross correlation between the channels in the RGB color space and the cross correlation between the channels of the HSL color space. Due to the strong similarities between HSL and HSV, an investigation of the cross correlation in HSV was deemed redundant and the results from HSL would represent them both.

The results found in table 5.1 clearly shows that the cross correlation between RGB channels is much higher than the cross correlation between HSL channels. The RGB channels all have a cross correlation coefficient above 0.99 (The maximum possible is 1). The cross correlation is much lower for HSL. Between Hue and Lightness the average lie at 0.25, between Hue and Saturation 0.42 and between Saturation and Lightness 0.75. These results are what we expected. Since RGB is a additive color space, all channels are dependent on the light intensity. They all respond similarly to fluctuations in lighting conditions during recording. On the other hand, in HSL only Saturation and Lightness are dependent on light intensity, while Hue is not. This shows with the higher coefficient between Saturation and Lightness than the other coefficients. Saturation is the only component dependent on both intensity and color tone. This is evident with Saturations higher coefficient with the two other components then

between Hue and Lightness. Hue and Lightness are dependent only on color tone and light intensity respectably, and have therefore a low cross correlation coefficient. But it is still high enough to indicate that there is some "leakage" between channels. It is hard to perfectly separate color tone and intensity from a picture, so there will always be some correlation between the two.

6.1.2 Visual interpretation of ICA on HSL color space

In figure 5.1 and 5.2 one can observe a sample of video taken in HSL color space with and without ICA respectively.

In the top figure in the non-ICA figure one can observe the Hue channel, and with visual inspection one can clearly observe what appears to be a HR signal. One can also observe some high frequency noise. In the Saturation channel one can observe what also seems like a HR signal, but at a lower amplitude than in the Hue channel. It also appear to have less general noise than Hue. The Lightness channel could have some HR signal, but it appears to be too faint to draw any conclusions.

If one compares the ICA figure with the non-ICA figure, one can observe that the Hue channel still boasts what appears to be the strongest HR signal. A lot of the high frequency noise seems to have been removed. At the same time the amplitude of the HR signal seems to be attenuated. We would still claim that the HR signal appears to be clearer after ICA.

In the Saturation channel it appears that the HR signal that appeared in the non-ICA figure have been attenuated, but they are still visible in some areas. In the Lightness channel some high frequency noise has been induced, but still no sign of a HR signals appears evident. When observing the general waveforms, it appears that only the Hue channel has retained its general waveform after ICA. The two other channels are unrecognizable from their previous channel. Mean channel value also seems to change an arbitrary amount after ICA.

In figure 5.3 one can observe a FFT of the Green and Hue channel before and after FFT. Here it is evident that we have a strong frequency peak at the HR frequency. It appears that the Hue channel have less energy outside this frequency peak than Green has.

ICA does not appear to change the frequency spectra much for either channel. The high frequency noise reduction found in the trace is outside the HR frequency range so it does not seem to affect the signal from a HR detection perspective. When observing energy outside the main HR frequency peak, ICA does not seem to affect the signal significantly.

From a purely visual inspection of this single ideal case video sample, it appears that the strongest HR signal is to be found in the Hue channel, with some of traces of it in the Saturation channel. The Lightness channel does not seem to have evidence of a HR signal.

6.1.3 Statistical results of ICA

In this section we will discuss the results found in results section 5.2.2, 5.2.3 and 5.2.4.

In the works by Poh et. al[23], when they used video quality comparative with our realistic case video quality measurement, ICA improved the results a fair amount in the Green RGB channel. Meanwhile in the works by Rustand

[28], ICA did not appear to improve results significantly. In contrary it could in some cases actually downgrade the results. We will investigate how our results present themselves in reference to these opposing conclusions. We will also discuss the effect of ICA on HSL and HSV color spaces.

In table 5.2 one can observe the results of using ICA on higher quality video. From the statistical data in this table there is a strong indication that ICA does not improve results on higher quality video. On the contrary it appears to reduce the success rate in some channels. If we observe the different color spaces individually, the RGB color space, here represented with the Green channel, has a small decrease of 2% success rate when using ICA. This is in line with the results from [28]. In the HSL color space, ICA improves performance in the Saturation and Lightness channels, but decreases it in the Hue channel. The same pattern emerges in the HSV color space. The improvement is the largest in the Lightness/Value channels. This is not surprising since these are the noisiest channels. The best ICA results are found in the ICA Saturation channel, which has a success rate of 63 % in both color spaces. This is approximately the same success rate as can be found in the best non-ICA channel. This indicates that ICA does not generate an overall *improvement* of results when using high quality video, with the perspective of using this in functional non-contact PPG system.

In table 5.3 one can observe the results when using lower quality type video. The results here indicates that ICA does improve results more in the low quality range than with higher quality video. The biggest improvement can be found in the RGB color space. The Green channel had large improvement in success rate, which went from 20% without ICA to 66% with ICA. These findings are in accordance with the findings in [23]. In the HSL and HSV color space, ICA improves results in the Value and Lightness channels, but worsen them in the Hue and Saturation channels. The best ICA channel is not better then the best non-ICA channel, so the use of ICA does not appear to improve the success rate when using lower quality video, with the perspective of using this in functional non-contact PPG system..

There are not enough samples to accurately compare the different channels against each other. This was not the purpose of this experiment. The purpose of this experiment was to determine if ICA might pose an improvement with regards to HR detection compared to normal color space data. We feel we have sufficient samples to establish a pattern to evaluate this in this experiment. Comparing different color spaces against each other was performed in results chapter 5.3 and will be discussed in chapter 6.3.

In summary it appears that the HSL and HSV color spaces are not more suitable for ICA than RBG color space. This in spite of the channels in HSV/HSL being more statistically independent than the channels of RGB. One could argue that perhaps the application of ICA on HSV/HSL was better than its use on RGB in itself, but that the results from HSL/HSV are so good in reference to measuring HR, that this surpasses the improvements made by ICA in separating the different sources.

For example the ICA algorithm had a tendency to shift the order of outputted sources, in some instances could the best HR signal be found in the outputted source 1, and in other instances it could be found in source 2. It is perhaps inaccurate to describe a channel as “HSL Hue with ICA” when the ICA algorithm does not output a channel that describes the hue of a frame after

transformation. It rather tries to separate the different sources of the signals into different channels. As such the outputted channels from ICA should perhaps more correctly be referred to as source 1, 2, 3 etc. However we have chosen the labeling we have for simplicity.

With regard to the shifting of channels that outputs the HR source, one could argue that we should have only used the channel which outputs the correct HR source as a basis for calculating success rate and the efficiency of detecting a HR signal. Now since the HR source has a tendency to shift channels, it will dilute the success rate of each individual channel. One could for instance claim hypothetically that *one* of the ICA channels detects a correct HR signal in 90% of the cases, and that therefore ICA is an improvement to this system. But to determine which channel outputs the "correct" HR source signal, requires prior knowledge of the HR signal, which one does not have in a realistic situation. Therefore arguing that *one* of the channels is more efficient then the best color space channel is impossible since we do not know which *one* channel that is.

In future works one might implement a variant of ICA or BSS that locks a specific source to one output channel, but that is beyond the scope of this paper.

6.2 Medical experiments related to PPG

In this under section we will discuss the results of the medical experiments related to non-contact PPG.

6.2.1 Detecting PPG signal from the feet

In other papers on the subject of non-contact PPG, to our knowledge, only the regions of the forehead and the palm of the hand have been used to detect PPG signals. We investigated if other regions of the body could be used.

In results chapter 5.4 it was investigated whether it was possible to detect PPG signals from ROIs placed on the top and bottom of the feet. In figure 5.6 one can observe two FFTs from two recording, with ROIs placed under the feet. One can clearly observe a clear frequency peak in the correct frequency range for the HR in both figures. The frequency peaks are also consistent on both the frequency spectra of both the left and right feet, indicating that these peaks are not the result of noise.

These spectra also shows a variation, meaning that there are several frequency peaks along the spectra, consistent with the frequencies that the HR varied to during the experiments. It shows the difficulty that is caused by heart rate variability. For instance, in 5.6(a) the HR went from 58 to 67 during recording. This reflects on the frequency spectra, with a peak at 60-61 bpm and a bigger one at 67. If the HR had been consistent at 67 bmp. during the entire recording, the frequency peak at 67 bpm. would have been stronger. Instead, the signal energy is divided between the two frequency ranges.

If one observes the spectrum of the left foot in figure 5.6(a), the heart rate variability causes the low frequency noise found at 55 bpm. to actually have a higher frequency peak then the PPG HR signal, although the PPG HR signal is clearly detected during the recording. Using a simple peak detection within HR range as the selector for HR, as we have used in the C++ program for this

thesis, this spectrum would have evaluated the HR to 55 bpm, and thus it would have given an error in the statistics.

The results found in these spectra is a strong indicator that the underside of the feet can be used to detect PPG signals. This results confirms that we are able to use the underside of the feet as a measuring point in the phase difference experiments. At the same time the PPG HR signal seems to be weaker than that found on the forehead.

In figure 5.8 one can observe two FFTs from two recordings, with ROIs placed on top of the feet. On contrary to the ROIs placed on the underside of the feet, these FFT shows no signs of a frequency peak around the HR frequency. In 5.8(b) the frequency spectrum of the left foot have a small peak where the HR lies, but this is not consistent with the spectrum of the right foot. All the other spectra only show signs of random noise, and no discernible frequency peaks that could be the result of a HR signal.

This result indicates that the top of the feet is not a good measuring point for PPG signals. As a result, this point was not used as a measuring point during the phase difference experiments.

6.2.2 Detecting PPG signals from darker skin tones

In result section 5.5 it was investigate if the skin tone of the subject had any effect on the PPG signal.

In figure 5.10 one can observe the frequency spectra of three 1000 frames long recordings of a subject with dark (non-caucasian) skin tone. The spectra shows no signs that dark skin tone have any negative effect on PPG reading. There is a very clear frequency peak around the HR frequency on all three spectra in most of the color channels.

If one observes the the spectra of the different color channels, some defining characteristics of the different channels appear. First, in very high quality video with a controlled, stable environment, the Green channel appears to have to have the clearest frequency peak at the HR frequency. It is closely followed by the two Saturation channels from HSL and HSV. This is because these channels are, as described earlier, dependent on both light intensity and color tone. Since the PPG signal is created by the changing blood flow in the capillaries, this effects both the intensity of light being reflected (by the increase in blood, absorbing more of the light during a "beat") and the changing of color tone in the skin. The Saturation and Green channels are able to pick up on both these signal components, and thus they have a clear frequency peak at the HR frequency peak. On the other side, the Hue channel only reacts to the change in color tone. If one observers figure 5.10(a) one can observe that, although the Hue channel has the strongest frequency peak at the HR frequency, a lot of energy in the signal lies outside the HR frequency peak. One can observe that the HR frequency peak is only slightly higher than the surrounding noise in contrast to Saturation and Green channels. This can indicate that Hue is more susceptible errors due to the noise created by σ_w and σ_g than Green and Saturation. On the other hand, it should be more resistive to noise due to σ_m than, due to this noise source mainly affecting the light intensity characteristics of the image.

The Value and Lightness channels do not appear to detect a strong HR signal. We do not thus expect these channels to yield much information to our use.

6.2.3 Phase difference

In this section we will evaluate the results of the phase difference experiments.

Experiment 1

In results section 5.6.1 one can observe the results of experiment 1. This was an experiment to test whether it was possible to get valuable phase information from the PPG signal, and if it were concurrency between the phase of the right and left side of the body without occlusion on either side. In this test it was found that the Green channel gave the best results.

In figure 5.12 one can observe the FFT of the experiment. One can observe a clear frequency peak at the HR frequency for the left hand, and a smaller frequency peak for the right. But at the same time there is a strong frequency peak for the right hand at 60 bpm, indicating that some noise or disturbance in the recording is present for the data from the right hand. In figure 5.13 one can observe a bandpass filtered trace of the Green channel. The left arm presents a quite good, consistent result with stable amplitude (except for the occasional spike). The right hand is slightly more erratic with occasional double peaks. Comparing the phase of them does not give a consistent result, with the right hand apparently moving in and out phase with the left, it is sometimes lagging and sometimes leading on the left. It appears that noise has obscured the results of this experiment too much to give any definitive indications. This experiment is an example of how vulnerable non-contact PPG systems are to noise.

Experiment 2

In results section 5.6.2 one can observe the results of experiment 2. This experiment was design to investigate the effects of occlusion on the phase difference.

The results were divided into three parts, in the first the right arm was fully occluded. In the second the full occlusion was removed halfway through and in the third part there was no occlusion.

Frequency spectra for both arms from part 1 is observable in figure 5.15. One can observe a distinct frequency peak at the HR frequency (77 bpm.) for the left arm. This peak is missing in the right arm which is occluded, there is a frequency component rising to small maximum at 110 bpm, but no distinct peaks at any frequency. In the trace in figure 5.18 one can observe that the left arm has a regular cycle with a mostly consistent amplitude. The right arm has a lower amplitude and is more irregular. The strong fluctuations at 2-5 seconds are probably attributed to movement. Regarding phase difference it is hard to observe any consistent patterns. What was hoped to be observed was a consistent lagging phase from the occluded arm. The local minima is our best measuring points, but lacking any consistent patterns, it is hard to make any definitive conclusions. It appears that due to the reduced flow of blood into the occluded arm, the PPG HR signal has been to reduced by such a degree that it is indistinguishable from random noise.

In part 2, the occlusion is removed halfway through. The frequency spectra of both arms are observable in figure 5.16. The frequency peak at HR frequency is still observable in the left hand. The right hand has a small peak at this frequency, but the surrounding frequency peaks are stronger. For the right arm

there does not appear to be any distinct frequency peaks, and the spectrum is mostly flat. But due to the abrupt change in situation, this is not unexpected.

In the trace of the Green channel in figure 5.19, a change before, during and after release is clearly visible for the right arm. The left hand has a consistent and stable pattern, and it is reasonable to assume that the pattern observed is the result of a HR signal. The signal from the right arm before release (0-14 seconds into trace) is more stable then the signal found from the right hand in part 1, but is still quite weak and it is uncertain if the signal is just the result of random noise. Most minima from the right arm do appear to lag behind the minima of the left arm to some degree in the period, but the signal is to weak and erratic for us to draw any definite data from with regards to phase difference.

In the period right before and during release (14-19 seconds) there were some disturbances in the recording due to the removal of the medical cuff which was creating the occlusion, so results from this period should viewed with some suspicion. However, one can observe that signal amplitude from the right arm increases after release as the blood rush back into the limb.

In the period after release (20-32 seconds) we have a stable signal from both limbs. The amplitude of the right signal is still smaller then the left, but this could be an after effect of the long occlusion. The signals appear to oscillate mostly in synchronization with each other.

In part 3 we observe both arms in a period after the occlusion has been removed. The frequency spectra in figure 5.17 show clear frequency peaks at the HR frequency for both limbs. The frequency peak for the recently occluded arm (right) is stronger and clearer then the other arm. In figure 5.20 the trace show a strong and stable signal from both arms. They both concur with each other and the HR frequency, so it is reasonable to assume that this is a HR signal. This is the result we hoped to achieve in experiment 1. One can observe that with such a stable result, there are still some fluctuations in phase between the signals, it is not a large fluctuation (less then 0.15 sec.), but it is still present and can be observed. There should ideally be no phase difference between the signals, due to the measuring points being at a similar distance from the heart. We believe that these fluctuations are due to random noise. These fluctuation show us that we should be careful making any definitive conclusions regarding phase difference results.

Traces in other color spaces were performed, notably Hue and Saturation, and they yielded similar results. But the best results regarding phase difference was still found in the Green channel.

Experiment 3

In results section 5.6.3 one can observe the results from experiment 3. In this experiment the effects of gradually increasing the occlusion was investigated.

The results were divided into 3 parts.

In part 1 the medical cuff was not turned on and there is no occlusion. FFT can be found in figure 5.22 and the trace in figure 5.25. The frequency spectra of the left arm has a reasonable peak at the HR frequency, however a lot of the energy resides outside this frequency. The right hand has a single frequency spike, but it resides outside the HR, at approx. 70-71 bpm. The frequency resolution for this sample was 2.52 bpm, so the discrepancy is probably caused

by noise. The trace of the Green channels shows irregularities between the right and the left hand, with no regular oscillation patterns. To much noise obscured results, and therefore no definitive conclusions can be drawn from part 1.

In part 2 the medical cuff is gradually tightened and thus the occlusion of the right arm is gradually increased. FFT can be found in figure 5.23 and trace in figure 5.26. The FFT shows a good frequency spike at 73.5 bpm, the frequency resolution for this sample is 1.5 bpm. so it is just within what we would call acceptable margins of a successful detection. The frequency spectrum for the right arm show a fascinating pattern, with three consecutive dropping frequency peaks, with the first and highest peak at 75 bpm, at the HR frequency, the next two peaks are at 78 and 81 bpm, with an equal increase of 3 bpm. for each of them. In the appendix figure A.10, which shows the results from the same stage in an identical experiment, shows signs of the same pattern, although it is much fainter and not as regular here.

The trace shows periods with a lot of noise. The left hand has a quite regular oscillation pattern while the right is a lot more irregular. From 0 to 11 seconds, there does not appear to be any regular pattern of phase delay. But from 11 to 20 seconds there seems to be a regular pattern emerging, with the local minima of the wave of the left arm is consistently leading on the right arms. The delay is slightly irregular, but still present. The results from this time section were good enough that we will return to these results for further analasys a little later in this section.

From 21 seconds and to the end of the trace, the waves seems to be in synch again. At 30 seconds the occlusion is removed, which could explain the spike seen 32 seconds into the recording.

In part 3 the there is no occlusion, and we try to investigate the unoccluded, or baseline, results again. The FFT can be observed in figure 5.24 and the trace in figure 5.27. The FFT shows frequency spikes at the HR frequency for both hands. The spectra also shows a spike outside the HR frequency for the right arm, indication some noise in the trace.

The trace shows a stable oscillation for the left arm, but some disturbance and irregularity in the right. One can observe the spike in the right arm at 4 seconds into the recording, possibly showing the results of the release of occlusion. The waves seems to move mostly in synch but with some irregularity as is present in most of the results.

Phase difference in feet

This experiment found in result section 5.6.4 is similar to experiments 2, except that we here investigate if we can detect phase difference between the feet, in contrary to the hands as usually done.

This experiment is divided into three parts. In part 1 and 2 the right foot is occluded, while the occlusion is removed between parts 2 and 3.

FFT for part 1 can be observed in figure 5.28. It shows no frequency peak for the occluded foot at the HR frequency. For the unoccluded foot there is a peak at the HR frequency, but a lot of the energy resides outside this frequency. This could either be due to the varying HR during the experiment or the present of strong noise. The trace can be found in figure 5.31. As the FFT predicted, there is a lot of noise and not a lot of information can be obtained from the trace. No discernible phase difference can be extracted from the two waves.

The FFT of part 2 can be observed in figure 5.29. The left foot shows a peak at the HR frequency, but at the same time a strong noise component at lower frequencies. The right foot shows no strong frequency peaks. The trace can be observed in figure 5.32. The amplitude of the right foots oscillations weaker than that of the left foot. The left foot also shows some periods with higher amplitude than the rest, but these are probably due to movement. For most of the experiment, oscillations are too weak and erratic for the right foot for us to draw any conclusion regarding phase difference between the feet. The left foot seems to have a more stable pattern then the right, but this is not surprising due to the stronger blood flow to the left foot.

The FFT of part 3 can be observed in figure 5.30 and shows a strong frequency peak for both feat at HR frequency. The trace observed in figure 5.33 shows two good and stable waves. No distinct phase differences is present between the waves.

This experiment once again shows that there is a correlation between occlusion and the radiated PPG signal. In the first two parts the PPG signal is much weaker and consists mostly of noise, while the last part shows a strong signal, were both feet are in phase. The difference between parts 1-2 to part 3 is distinct. Unfortunately the signal is to reduced because of the occlusion, that any phase information is obscured by the underlying noise floor. It is speculated that the occlusion used in most of these experiments were too strong and that for future experiments, a weaker occlusion is used.

Sources of error in these experiments

In the experiments in this section, there have been some problems with noise, and especially σ_m, this is probably caused by the occlusion making it harder for the subjects to keep the arm still. When the medical cuff expanded, it at the same time pushed on the elbow joint. Because of this the position of the arm moved in accordance with the inflation. The subjects at the same time had to use muscle strength to try hold the arm in the same position. Holding an position static against a force, causes the muscle to twitch and shake after a while (if holding a heavy object at arms length for instance, ones arms will start to shake shortly). These disturbances were only slight, and were not detected during recording. This caused many of the attempts to be discarded after being analyzed. Most of the experiments were performed numerous time, and the best results have been presented here.

Other error sources causing σ_m are the physical motion of heart, propagating through the body during a beat. These motions are only slight in the extremities, but they are at the exact same frequency as the HR, and the disturbance caused by this could be hard to distinguish from a true PPG HR signal.

The use of the bandpass filter could also give errors. The low frequency noise usually ends at 40-50 bpm (0.66 - 0.83 Hz) while the resting frequency of the HR is usually found at around 60 bpm. This gives us a very narrow transitband for filtration between highest frequency noise in the low frequency range, and signal we are trying to detect. We decided to have a stopband frequency at 45 bpm. with attenuation of 60 db while the passband was set at 60 bpm. This gives us a transitband of only 15 bpm (0.25 Hz) for a 60 db transition. Accordingly we need a very steep filter with many coefficients. This can cause a problem with transients occurring after filtration and phase delay. Using a zero phase filter

(filtrating the signal forward and backwards) removed the problem with phase delay. A FIR filter with 63 coefficients was found to give the best results. Some IIR filters were tested but they usually gave larger transients. The transients induced by the filter could in some instances be mistaken as a PPG signal.

These experiments were filmed with a Sony PMW-EX1 camera. This is a professional grade camera of high quality. This camera was used because it was perceived that with its improved optics it could produce a better results than a ordinary camera. Unfortunately this camera stores files in a compressed XDCAM codec which require conversion to another format before use. It is supspected that this compression and conversion destroyed some of the color information in the video. This could explain why we got worse results in the HSL and HSV color spaces than when we use a webcam that does not need a transform before use. The bad results in HSL and HSV required us to use the inferior RGB color space which might explain some of the noise.

Results from experiments 3 part 2

The results from experiment 3, part 2, were interesting enough to incite further analysis, especially the period from 10-20 seconds into the experiment. If one observes this section, one can observe that there is a regular difference between the local minima of the left and right arm, with the right, occluded arm, lagging behind the left. This goes in accordance with our theory about measuring increase in PTT when an occlusion is present.

In table 6.1 one can observe the estimated for phase delay between the extremities taken from trace 5.26. Estimating from 11 to 20 seconds in to the trace. The data in the table show a consistent delay between the right an left arm. At only one point (18.5 seconds into trace) does the right lead on the left arm. The delay between the arms has a mean value of 0.2036 seconds, there does not seem to be an increase in delay with time, but this can be caused by the medical cuff inflating stepwise i.e. it inflated to a certain level and held it there for a time period, then increased again. Through there are variations in the phase delay, we have not encountered any such consistencies over a period. In previous experiments, where the trace was dominated by noise, the noise seem to vary more and have less consistency. We can therefore make the assumption, with reasonable certainty, that the phase delay observed is not only the result of random noise and can therefore be used for further calculations.

After the 20 second mark the waves seem to synchronize again and there does not seem to be a consistent delay. This could be caused by the occlusion going beyond a certain tipping point, where the flow of blood is so reduced that other signal sources than the HR is the dominant signal in this frequency domain. This could be the reason why we did not get any useful data from the full occlusion parts of experiment 2.

To get a stable reference point to measure PTT against, we have in figure 6.1 synchronized figure 5.26 with ECG taken at the same time. With reference to the ECG, the delay between the ECG peak and the local minima of the right and left appears relative stable. If we chose the top of the R wave as a reference point and measure the time delay to the next local minima of the right and left wave we get the PTT to each extremity. We have measured the PTT within

Time into trace[s]	Delay, right hand relative to left[s]
11	0.36
12	0.16
13	0.24
13.7	0.36
14.5	0.08
15.5	0.32
16.2	0.20
17	0.28
18	0.16
18.5	-0.08
19.5	0.16

Table 6.1 – Estimated delay between left and right hand. Approximated from trace in figure 5.26. Experiment 3, part 2.

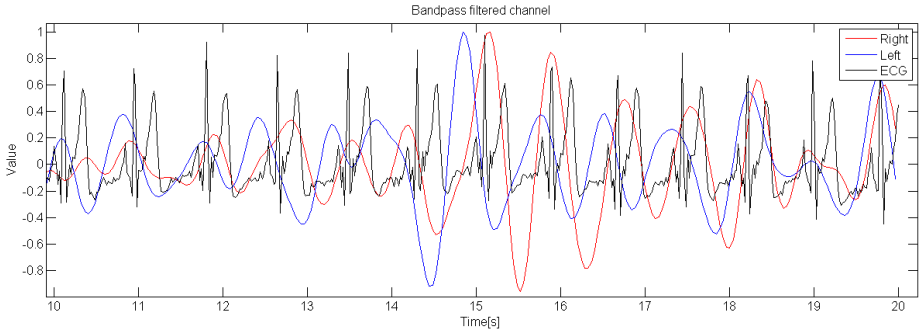


Figure 6.1 – ECG synchronized with segment of figure 5.26. Taken from experiment 3, part 2.

the time period 11-20 seconds. The results can be observed in table 6.2.

With this information it is possible to estimate the blood pressure below the occlusion by using equation 2.4 from theory chapter 2.1.4.

$$P_e = P_b - \frac{2}{\gamma PTT_b}\Delta PTT$$

The blood pressure at this period was measured at 118/78 mmHg. If one use systolic pressure we get P_b=118 mmHG. We make the assumption that the PTT of left arm, equals the PTT of the right arm if it was unoccluded, therefore we use $PTT_b = 0.25$ s. Using average values for the time period we get $\Delta PPT = 0.46 - 0.25 = 0.21$ s.

Using mid value for $\gamma = 0.017$ mmHg^{-1}

$$P_e = 118 - \frac{2}{0.017 \cdot 0.25} \cdot 0.21 = 19.17\, mmHg$$

This is not a unreasonable pressure, for a case of strong, but not complete occlusion. In this case the blood pressure was low, but high enough for some blood to flow. This can also explain why we did not get any usable phase delay results after the 20 second mark of the trace. If the occlusion increased, it is not

Time into trace[s]	PTT Left arm[s]	PTT Right arm[s]
11	0.26	0.62
12	0.28	0.44
13	0.32	0.56
13.7	0.04	0.4
14.5	0.14	0.22
15.5	0.10	0.42
16.2	0.22	0.42
17	0.16	0.44
18	0.40	0.56
18.5	0.50	0.42
19.5	0.38	0.54
Average	**0.25**	**0.46**

Table 6.2 – Time delay to next local minima of the right and left arm with reference to the R wave

unreasonable to assume that the pressure was not high enough to push blood in to the capillarity closes to the skin and we would therefore not detect a HR signal.

It must be said that this calculation was done only on a single subject and on a single sample, and it is therefor not sufficient for us to be able to draw any conclusive statement on this method. But the results are in line with the results found in [18]. They used a method of *contact* PPG in contrary to our *non-contact* PPG, and they were therefore able to avoid many of the noise sources we have encountered.

All in all this experiment shows the potential viability of this method in calculating blood pressure by a non-invasive method. These results also validates further research into such a method. If one in future works were able to get a measurement without many of the noise sources we have observed here, we expect that a increased success with this method could be found.

6.2.4 Synchronization of ECG and non-contact PPG signal

In this section we will review the results shown in results section 5.7. In this section we investigated the correlation between the ECG of a subject and the detected non-contact PPG signal from the skin.

Synchronization without occlusion

We first review the results shown in the trace shown in figure 5.34 from results section 5.7.1. The trace shows two signals. One is the ECG and the second is the detected PPG signal, bandpass filtered in the HR range.

The noise in the first section (0-4 s) is due to a synchronization pulse that was visible in both sources. There is some noise from 8 to 11 seconds into the recording, and this is due to the subject turning his head in this period. This gives us a valuable opportunity to observe the transient response of the FIR bandpass filter. The subject used approximately 2 seconds to move his head 45 degrees, meaning that the signal disturbances in the section from 8-10 seconds are due to the movement itself and the disturbances from 10-11 seconds

are due to transient response of the filter. What can be observed is that only the amplitude of the next wave seem to affected by the transient. The waves more then a second later are not effected by the pulse. If one observes the first local minima after the noise pulse (approximately at the 11 second mark), it is not out of phase with the consecutive local minima, meaning that the phase information is not distorted by the filter, indicating that the zero phase filter works as intended. There are also noisy sections later in the trace (at 38, 50 and 63 seconds into the trace) were the oscillation pattern changes slightly in reference to the surrounding waves. This is mostly visible through a sudden increase in amplitude. These sudden changes in amplitude are probably not due to any biological factors. There could be an unknown or unaccounted for biological factor affecting the radiated PPG signal, such as a sudden dilation or contraction of the blood vessels nearest the skin, but it is thought that this would only induce a minor and more gradual change in the signal. Probably these disturbances are due to some form of noise. If one observes the local minima during and following the noisy section, they do not appear to move much relative to the ECG, indicating that the phase information in signal stays intact when the noise pulses are small.

If we take the top of the R wave as a reference point and set it as t=0, we can obtain the PTT by measuring time delay to the next consecutive local minima of the PPG signal. From the data in figure 5.34 we get a mean PTT of 0.3342 seconds with a standard deviation of 0.0462 seconds. We measured a PTT of 0.25 seconds to the palm of the hand without occlusion in a previous section. A larger PTT to a part of the body that appear closer to the heart than the palm of the hand might appear illogical, but such things as distance to the closest arteries will influence PTT, and the palm of the hand lies right on top of the radial and ulnar arteries. The measurement of PTT to the arms was also performed under noisier conditions and with fewer samples, and are therefore more uncertain.

Synchronization with gradual increase in occlusion

In this section we will review the results shown in figure 5.35 from results section 5.7.2. In this experiment the occlusion was gradually inflated in the hopes of achieving a noticeable change in ECG, and compare how that affected the PPG signal. Unfortunately (but probably fortunately for the subject) the occlusion does not appear to have effected the ECG to any noticeable degree. But due to the increased movement and disturbances from the use of a medical cuff in this recording, relative to the experiment without occlusion, we can investigate how a noisier situations affects the synchronization.

This recording was performed later then the previous experiment in this section, this has affected the synchronization between the ECG and the PPG signal because of clock drift after synchronization. The effect of this is that the timing has been shifted between the two signals by approx. 0.06 to 0.08 seconds with reference to each other, with the PPG signal is lagging behind the ECG. This is noticeable by the first local minima happening earlier with reference to the R peak then in the previous experiment. There is a chance that this decrease in PTT could be a effect of the occlusion. It may be that the occlusion in one extremity could increase the blood pressure in the rest of the body, thus decreasing the PTT to the rest of the body. But we consider it more likely that

this perceived decrease in PTT is due to clock drift. If this experiment is to be repeated, and one gets the possibility to eliminate potential synchronization errors, it would be interesting to observe if this pattern of decrease in PTT because of occlusion in a extremity repeats itself.

Because of the increased disturbances in this recording, the number of and severity of disturbances in the graph increases. This is most noticeable at the 13, 30 and 37 second mark. These disturbances sometimes causes the PPG signal to miss a heartbeat (13 and 37 second mark) and in some cases create a extra peak where there is no heartbeat (30 second mark). These errors would have been impossible to detect without the extra information given by ECG and would in an automatic non-contact PPG HR system cause a measuring error. But with only 3 errors during a 80 second recording, it is within what we would call an acceptable margin of error.

Synchronization with sudden removal of occlusion

In this section we will review the results shown in figure 5.36 from results section 5.7.3.

In this experiment we had a full occlusion of the right arm, which was suddenly released, in the hopes of observing some change in the ECG that we could then relate to change in the PPG signal. The occlusion was removed at the 47 second mark. No discernible change in the ECG are observed after the occlusion is removed. We do not either observe any change in the PPG signal.

This experiment also suffers from clock drift, like the previous experiment in this section did. The drift has now increased to approx. 0.08-0.1 seconds, with the PPG signal lagging behind the ECG. If the PTT was changed due to the occlusion, we should have seen a change in the period after the occlusion had been removed, and the PTT would have returned to the value we measured in the first experiment in this section.

The recording was performed under similar conditions to the previous experiment, with some noise and disturbances due to the removal of the occlusion on the subject. This has created some areas with disruption, but only in one place could the disturbance potentially cause an error in an automatic non-contact PPG system (35 second mark).

Summary of ECG and PPG synchronization experiment

These experiments have given useful results and insights into non-contact PPG systems. We will draw the conclusion that there is a strong correlation between the arterial blood pressure (ABP), which is synchronized to the ECG, and the PPG signal. If the PPG signal we detected had been the result of another biological process in the skin or just random fluctuations due to noise in video recordings, there would have been a lot more random fluctuations, false detections and missed detections. We also observe that there is a high degree of synchronization, but with a delay, between the ECG and PPG signal.

These experiments used a completely healthy subject, and we did not expect to get any noticeable changes in the ECG induced by the occlusion. If these experiments were repeated in any future works, it would be interesting if the subject had a form of heart rate arrhythmia, this to investigate how arrhythmia in ECG affects the radiated PPG signal. In these experiments, there were no short term or drastic changes in the ECG, as should be expected when using a

healthy subject. But the strong correlation between ECG, and by an extension ABP, and the emitted PPG signal suggest that we should be able detect an arrhythmia. The challenge is how separate noise from an actual arrhythmia. There would be a primary focus on creating an as stable as possible environment for recording, where the chance for noise is minimized.

If detection of arrhythmia can be done by a non-contact PPG system, by cheap, consumer grade video equipment, it could lead to large improvement in detection and treatment. Today, heart and circulatory related illness is the most common cause of premature death in Norway and the developed world[21]. Arrhythmia is a god indicator for an underlying medical problem, and in many cases, early detection and treatment or arrhythmia could save lives. Unfortunately, detection of an arrhythmia is expensive and time consuming. In many cases, a patient must be connected to an ECG recorder for an extended period of time in the case an arrhythmia occurs. And often such measurements take place after an underlying medical condition has revealed itself, which in some cases is to late. If early detection of arrhythmia was cheaply and easily available for the general public, many more individuals with arrhythmia could be detected before it was too late. Such a cheap and easily available detection system could be possible with the use of a non-contact PPG system, using a smartphone or a laptop with a webcam.

6.2.5 Phase difference in laparoscopy and colonoscopy video

In this section we will review the results found in results section 5.8.

The main purpose of this experiment, was if we could observe any noticeable differences from the ROI holding a cancerous growth to other ROIs holding normal healthy cells. There were four ROIs extracted from a single, 7 second clip, showing a cancerous growth in the liver. The cancerous growth was present in ROI 1 while the other three ROIs were used as comparison.

The frequency spectra of the different ROIs can be observed in figure 5.38. The spectra shows no discernible differences for the cancerous growth relative to the other ROIs. There is a slight frequency peak for the cancerous ROI at 2.4 Hz which is not present in the other ROIs, but we believe this is just caused by random noise. The segment is also to short to get a reasonable frequency resolution.

A trace of the Hue channel can be observed in figure 5.39. This graph does not either present any characteristic that separated the cancerous ROI from the other. Except for ROI 4, which does not appear to have a large blood flow, the cancerous ROI has a phase similar to at least one of the other ROIs inn all oscillations.

The recordings used here were not purposely recorded for PPG analysis, so they did not have the emphasis on stability that is required for such an analysis. If a longer, more stable recording were made, a better result may be possible.

A large problem with cancerous growth is that they are hard to separate from other healthy tissue by visual inspection alone. If one observes figure 5.37 there is no clear visual indications that ROI 1 contains a cancerous growth. This similarity with healthy tissue presents a challenge for surgeons and doctors, both during detection and surgical removal of cancer. If a automatic detection system of cancerous cells based on video is possible, it would be a very valuable tool for doctors.

6.3 Discussion of HR detection C++ program

In this section we will discuss the results given by the C++ framework developed for this thesis. We will discuss the found optimal parameters for the tracking algorithm and how different circumstances affected the success rate of the program. In subsection 6.3.1 we will discuss the characteristics of the different color spaces, which one was best suited for HR detecting and how different recording circumstances influenced results. In subsection 6.3.2 we will discuss optimal parameters and algorithm for the tracking and recognition software. In subsection 6.3.3 we will discuss the largest challenges for developing the concept of non-contact PPG into an functional and efficient product both for the home market and for medical applications.

6.3.1 Characteristics of the different color spaces

Discussion in this subsection will mainly refer to the results found in table 5.4 found in chapter 5.3, but we will also discuss some of the general characteristics found when using this system.

In the table one can observe the efficiency of the different channels with regards to detection HR. One can observe that the most efficient channel is the Saturation channel from HSV with a success rate of 57 %. Right behind follows HSL Saturation with a 53 % success rate and behind that again follows Hue with a 46 % success rate. The Green channel has a success rate of just 17 %. Value and Lightness has just a success rate of 7 % and 10 % respectively, but with the amount of noise observed in these channels, we did not expect any grater success.

It is apparent that the performance of the Green channel drops quickly with the decrease of quality. It is especially vulnerable to the effects of σ_m. On the other hand, the Hue and Saturation channels does not degrade as quickly and the probably of success with these channels are still reasonably good even at quite bad video quality. This goes in accordance with our theory that Hue and Saturation would be more robust in the low quality cases. The two Saturation channels often display the same result, this is not unreasonable since they are so similarly defined, but in some cases they differ. In the case of either Hue or Saturation fails, the other channel sometimes succeeds. We estimate that in at least 70 % of the cases, *at least one* of the channels succeeds. This could be useful if used in accordance with a statistical analysis of previous HR measurements. If the three channels all show different results, the channel which shows a frequency closest to the previous result probably is the most correct.

Comparison to results in other papers

Due to the different statistical measurements used in this thesis relative to other papers on the subject, it is hard to make a direct comparison between the different techniques in this and other papers. But we can make an approximation with the use of standard deviation. In the paper by Poh [23, p.12] they have a standard deviation of 4.59 with ICA. While in Rustand [28] there was a normal distribution of 2.1 bpm, in these papers only the Green channel from RGB color space was used. These are the results taken from the situations most similarly to the ones we used. If we assume a normal distribution of results, one can calculate the proportion of results that fell within 1.5 bpm. of pulse oximeter

measurements. ±1.5 bpm, from oximeter values was the interval for what we would usually measure as a successful detection. From this we calculated that Rustand got a success rate of 52 % while Poh got a success rate of 25.2 %. It must be noted that this is a somewhat crude approximation and the numbers should only serve as an indicator of our measurement metric of success rate. The measurements were also done under different circumstances and with different implementations then ours. In this thesis we have for instance not used any statistical analysis based on previous HR rate measurements and used this to ignore false positive detections (for instance a sudden jump in HR from one moment to another), we have rather focused on the efficiency of the individual color spaces without improvement.

From these results it is apparent that our success rate with the Green channel from RGB color space is lower then theirs. But our results from Saturation and Hue channels are greater. We expect that with the implementation of statistical analysis and a refined selection algorithm for frequency peak most likely to contain HR signal, we could see a great improvement in success rate using these two channels.

Observations related to efficiency of PPG systems

In this subsection we will discuss some observations made during development and testing of the system that affected the efficiency of the systems. These are not statistically proven observation and will therefore not be presented as a result, they are rather subjective observation made by the author that might help in a future development of a functional system.

One observation made was that the type of ambient light during testing affected the results. It was notices a discrepancy between measurements made on different times of the day, and this was traced back to if the measurements were made with daylight present. It turned out that when indirect sunlight (as during daytime, with sunlight shining through windows) was the dominant ambient light source, the efficiency of the Green channel was reduced, with reference to when the sun had set and indoor florescent lighting was the primary light source. Indoor light and sunlight have different frequency spectrum, with outdoor light having a lot more energy in the high frequency range, so it is not inconceivable that this can affect the how the PPG signal is reflected back from the skin. This observation validates further research into how the frequency spectrum of the light affects the reflected signal from the skin in future works. Different types of indoor lighting might prove itself better suited for measurements than other.

With reference to lighting situations, direct lighting without shadows on the skin produced a better result. When the recorded subject tilted his head slightly backwards, such that no parts of hair of forehead shadowed for the overhead lighting, we got a better result. Because of this we recommend placement of the video recording device higher than the subject eyes. This makes the subject unconsciously tilt his head backwards to look directly in the camera and one gets a clearer recording.

30 seconds is the longest period a subject can sit still, when window size increased much over this, errors caused by movement increased exponentially. We therefore do not recommend longer window size than this. It is also sufficient in most cases to get a reasonable frequency resolution at most frame rates. The best compromise between frequency resolution and the increased chance of noise

is found at recordings between 15-20 seconds.

With the exception of movement, the quality of video does not seem to affect our results noteworthy. Parameters like video resolution and the quality of the recording device does not seem to matter much. In this thesis two cameras were used. One Sony PMW-EX1 which is a 80 000 NOK production camera used by media professionals while the other was a 500 NOK Microsoft webcamera. The results from the high quality Sony camera was not better than that of the cheap webcamera. There were in fact some instances were the expensive camera had results that were worse than that of the webcamera. There was a slight difference in the results from the Saturation channel. This is probably due to that professional cameras usually store video files in a compressed format, and that this compression removes some of the color information. This compression probably offsets any gain we get from the improved optics of a high end camera. In the case of the webcam we can store the video in the format and quality we wish from the start. It would in future works be interesting to test if a professional camera storing in a uncompressed format such as RAW offers an improvement over a cheap webcamera.

6.3.2 Tracking and recognitions algorithms

The tracking software was implemented as described in the implementation chapter. There was two such algorithms implemented. One for detection of the face, and one for tracking features between consecutive frames to estimate motion. A combination of these two was found to give the best results. The face detection algorithm was used to detect the face and to place an ROI over the forehead. In Rustand [28] this was performed between each frame. This produced a workable, albeit "choppy" results. With choppy we mean that the coordinates of the ROI changed slightly with each frame, which slight differences between each individual frame, causing a lot of noise. Coordinate filtration alleviated the problem to some degree. In this thesis we investigated if a tracking algorithm based on tracking strong features could work better at tracking and compensating for movement. This algorithm based itself on finding a strong feature in picture, and measuring how much this feature had moved since the previous frame.

This solution worked in some cases better than Rustands solution, but it presented some new challenges. The biggest of these was what we call "drifting", short for when the ROI drifts away from its intended position and settles somewhere unwanted. The algorithm followed the strongest[19] feature in each frame. But what the algorithm chose as the strongest feature could suddenly change between frames, and the algorithm would believe the old feature had moved when it in reality had only found a new feature. This was alleviated to some degree by decreasing the strength of a feature as a function of distance from its last position, making the "old" feature most likely to selected as the strongest, althrough stronger features suddenly appeared in another position. To reduce "chopping" i.e. small unnecessary repositions of ROI between each frame, small changes in feature position did not change position of the ROI. The algorithm only updated the coordinates of the feature in this case. Only

[19]Strongest is a measurement created by the algorithm which shows how easy this feature is to separate from the background

when the feature had moved a sufficiently large enough amount from the original position did the ROI update its position.

The largest challenge for the tracking algorithm was the presence of hair in the frame. Hair presents itself as many very strong, but completely identical features close together. This made the algorithm misinterpret movement and the ROI would drift very quickly out of position and start to measure pointless hair instead. It proved itself completely impossible to use the tracking software when hair was within tracking boundaries. The algorithm is not able to understand and differentiate between different substances, so it is not possible to make it "ignore" unwanted tracking substances. To alleviate the problem only the area within the ROI was analyzed by the algorithm. The skin on the forehead itself is a largely homogenous surface with few strong features, so in some cases the algorithm was not able to find a suitable tracking point. This led to the algorithm not being able to compensate for movement of the subject. If the drifting was to severe, the ROI had no chance of returning to its intended position. To alleviate this problem a hybrid between the detection and tracking software was implemented. The detection software was used at intervals of 150 frames apart (approx. every 5 second) to detect the face and place ROI, while the tracking algorithm worked in between these intervals. This made the detection algorithm compensate for large movements, while the tracking algorithm compensated for the small movement. This proved to be a successful implementation and worked well, as long as the movement was not to excessive.

The largest problem was erroneous detection by the face detection software. The detection software worked quite well, with an success rate in detection the face very high, estimated at up to 97 % of the cases. But because of the increased number of detection per sample, the chance of *one* erroneous detection increased. With a normal window size of 500 frames, there were 4 detections per sample which means a 11 % chance of a false detection happening. The coffee maker seen in the background of figure 5.4 was a personal favorite of the detection software. Because of the suddenness of the change in the case of erroneous detection and change of ROI, the broadband noise caused the sample to be discarded. The tracking algorithm also made errors and sometimes drifted so far off target that the sample had to be discarded. This happened on an additional estimated 10 % of the cases. This means that around 20 % of the samples had to be discarded, which is a too high number for a functional non-contact PPG system. But we believe that with further improvement and tweaking of these algorithms, this error rate can be brought considerably down.

In future works, a lot of work can be made improving the tracking and detection software. There are lot of parameters that can be tweaked to fit individual circumstances. For instance can the face detection software be at closer intervals if there is a lot of motion. There are also many other feature tracking algorithms that might be more efficient at tracking features on the skin than the SURF algorithm used here in this thesis. A tracking algorithm based on the detection of other facial features such as the eyes and nose can be also be used to track motion. But this is beyond the scope of this thesis.

We believe that a hybrid system based on both the detection and tracking software, with a smart algorithm to combine them efficiently and to hinder errors, can utilize the inherent strengths of both methods to create a highly functional system for use in a functional non-contact PPG system deployed to smart phones, laptops and ordinary household consumer products with a

camera.

6.3.3 Challenges and future research

With regards to extracting a useful PPG signal from a video recording, the noise induced by movement, or σ_m as we have called it, is the biggest challenge. If one are to develop the system shown in this thesis in to a functional system for use by ordinary consumers, one must expect a large amount of "realistic" movement in the picture. To this cause, a highly functional motion compensation system must be developed. We believe that such systems have already been developed in several other instances and it is only the matter of converting such already existing systems to fit the requirement of a non-contact PPG system.

A smarter HR frequency selection algorithm should also be implemented. In this thesis we have based our selves on a simple peak detection within HR frequency range, this so we could focus on evaluating the performance of the different color spaces unaffected by later algorithms in the system. We expect that the overall success rate of a system could be greatly improved with such an algorithm based on both previous results, and the probability of different channels succeeding under different circumstances.

A challenge to this system is the real time requirement when using a webcam. In such a system, all analysis of an individual frame must be completed before the next frame is captured by the camera, or else we will "lose" the next frame waiting in the buffer, also known as a dropped frame. This means a maximum processing time per frame of 1/FPS. In a 30 FPS system this institutes a rather harsh maximum time available of 33.3 ms. per frame. This was one of the reasons this thesis was implemented in a C++ versus a Matlab environment, since, Matlab, being a much slower language, which could lead to more dropped frames if the algorithm does not complete on time. The most time consuming part of the system is the face recognition software, which uses an estimated 60 ms to complete detection of one frame in our C++ system. The rest of the system uses 20 ms combined, so the largest bottleneck is without doubt the recognition software. The added time usage of the recognition software means that we drop 1-2 frames each time the recognition algorithm runs. 1-2 dropped frames does not constitute a large problem, but if the time delay increases and the number of dropped frames increases much, it could soon become a problem. This software was run on a modern computer with a strong Intel i5 processor. While the processing power of a modern smartphone has increased a lot in the later years, it still cannot compete with that of a computer. We therefore expect the processing time on a smartphone to be some magnitudes larger than that of a computer. This will mean that without modifications, the system as is will drop a lot of frames. To combat this we can reduce the frame rate of the video, but this can only be done as long as the Nyquist rate is satisfied. Reduced frame rate also means more movement in between frames so that we get increased noise. If the frame rate can not be further reduced, one will have to find ways to decrease the time consumption of the system. This can be accomplished by either streamlining the code, or to use more efficient algorithms for detection. The drawback to this is that quicker algorithms are usually less accurate and prone to failures. The cascade classifier we have used is per now the most efficient, but we suspect it is to resource intensive for use on an mobile application. One could also investigate other tracking algorithms. A comparison

of the different available tracking algorithms can be found in [16]. The SURF algorithm used is not the fastest available, but it is one of the most accurate. One can for instance sacrifice some accuracy for decrease in processing time if that is a requirement.

Future app development

The framework has been developed with a future development of a mobile phone app in mind. We chose to use of the openCV framework partially because it has the possibility to be used both on the iOS and android platforms. These two are the two dominant platforms for smartphones at the moment. Most modern smartphones have the addition of a camera, and if one was able to develop a functional non-contact PPG system that used the on board camera, one could reach a very large market. This is positive both in a marketing and business perspective, but also from a public health perspective. If this technology for instance were able to detect the early signs of arrhythmia or other heart diseases, it could end up saving lives.

We believe that a mobile phone app could be relatively easily developed with a basis on this framework. There are some challenges one must overcome, but we believe these are not insurmountable.

7 Conclusion

In this thesis we have continued the investigations of non-contact PPG systems, and how the technology may be used and improved in the future. The theme of this thesis have consisted of two parts. The first is an investigation into how we can improve on the principle of non-contact PPG measurements of heart rate and to lay the groundwork for a future, functional system to be developed. The other theme is a general exploration of the capabilities of non-contact PPG measurements. We investigated what medical information was possible to gain from such information and how this potentially can help with medical treatment and diagnosis. To accomplish this, we have developed a framework in C++ using the openCV framework. This has helped us to investigate the wider concept of a functional non-contact PPG system.

7.1 ICA

The use of ICA, through the JADE implementation in matlab, has been investigated and tested on the RGB, HSL and HSV color spaces. It has been found to improve results on some channels, but in most channels worsening of results are observed. In the high quality video spectrum, the largest improvement was found in the HSL Saturation ICA channel. In this case ICA improved the channel results with 10 %. But the best ICA channel results were equal to, but not better than the best non-ICA channels. ICA does thus not make any improvement in the high quality regime.

In the low quality video spectra the largest improvement were found in the RGB color space where it improved the Green channel with 36 % up to a reasonable success rate of 56 %. Here the best ICA channel were 5 % less successful than the best non-ICA channel. In most cases ICA treatment actually worsened the results. Only in channels with very high amounts of noise did we find any improvement with ICA. ICA does not thus appear to make any improvement in the low quality regime either. ICA did not appear to function better with more independent color spaces, relative to the RGB color space, as we theorized. In the case one only used RGB color space, we can recommend the use of ICA in low quality situations. With the use of the more efficient HSL/HSV color spaces, the use of ICA does not serve any purpose with respect to HR detection.

7.2 The most efficient color spaces

We have tested the efficiency of different color spaces up against each other in realistic situations, and we have found that the Hue and Saturation channels of the HSL and HSV color spaces are far superior to Green from RGB in all situations with regards to HR detection. Without the use of improvement algorithms and based on a simple frequency peak detection, we have gotten a success rate of between 50-60 %. This is a large improvement relative to the success rate of the Green RGB channel which only had a success rate of 17 % in the same situation. The Hue and Saturation channels have proven themselves far more robust against noise and fluctuations during recording then the Green channel.

Only in one situation has the Green channel been shown to give better results. In high quality situations it appears to keep its phase information better than Saturation and Hue.

The Value and Lightness channels of HSV and HSL respectively do not appear to hold any valuable information regarding human vital signs.

7.3 Obtaining PPG signal from different regions and subjects with darker skin tones

We have investigated if it is possible to acquire a PPG signal from regions other than the forehead. We have investigated the regions of the palm of the hand and the top and bottom of the feet.

We were able to acquire signals from both the palms of the hand and the bottom of the feet, but we were not able to get a usable signal from the top of the feet. PPG signals from the hand and the feet were detectable, but much weaker and prone to noise than the forehead.

We observed no difference between the PPG signal from subjects with light skin tones and subjects with darker skin tones.

7.4 Phase difference in PPG signals

We have investigated if we were able to detect the phase of the PPG signal, and if we could use a potential phase difference between two point to: (1) Detect phase difference due to occlusion, (2) use this phase difference to calculate change in blood pressure.

The results show that we are able to detect the phase and to compare when measured the signals taken from the left and right palm. At the same time we have shown that these phase measurements are very vulnerable to noise. The best results were found in figure 5.20 where we had two synchronized signals over an extended period. With a full occlusion on an extremity, the PPG signals from that extremity became so weak that all information became obscured by noise. We were more fortunate with a partial occlusion where we in some samples were able to detect signs of a consistent phase difference. The best results were found in figure 5.26. We were able to, from these phase differences calculate a blood pressure below the occlusion point that was not unreasonable. Due to the extensive noise in these samples and few samples available for this reason, we are not able to draw any definitive conclusion on the subject of phase difference. But there are clear indications that there is a connection between occlusion and phase difference and that this subject needs to investigated further.

7.5 Synchronization between ECG and PPG signal

We have investigated if we could synchronize a PPG signal with a ECG signal and if we could obtain information from this. We wished to investigate if a PPG signal could be used to detect arrhythmia.

We succeeded in synchronizing the signals as can be observed in figures 5.34, 5.35 and 5.36.

There is a clear indication that there is a synchronization between ECG and the PPG signal radiated from the skin. With this we were able to measure a PTT from the heart to the forehead. A healthy subject had been used in these experiments, and we were not able to induce large enough change in the ECG as to be able to observe how rapid change in ECG, as with an arrhythmia, affects the radiated PPG signal. We suspect however, that due to the close

correlation between ECG and PPG shown, that one would be able to detect a change in ECG with PPG measurements. The real challenge lies in how to separate this change from noise. These results show a great potential and validates further research into the subject. If one were able to make a system that detects arrhythmia via video recordings, it would have a large positive impact on public health.

7.6 Phase difference in laparoscopy and colonoscopy video

We investigated if we were able to detect a phase difference between different tissue regions in both laparoscopy and colonoscopy video samples. In the first case it was with the purpose of detection necrosis after a lung transplant and in the second to detect cancerous growth.

Unfortunately were the videos too unstable for us to be able to draw any consistent measurements from them, and we were therefore unable to detect any differences or identifiable characteristics from the unwanted tissue.

It might be possible to detect a difference if one used video which was filmed with the purpose of a good non-contact PPG measurement. This should be further investigated future works.

7.7 A functional non-contact PPG system for consumer use

In this thesis we have developed a software framework in C++, which can be used as proof of concept of a functional non-contact PPG system being developed for consumer use. We have developed with the eventual future export to a smartphone system in mind. We have also tested different types of tracking and recognition systems to determine their best use.

Tracking system

With regards to tracking systems we have found that a hybrid system based on both face recognition using cascade classifiers, and the tracking point detection algorithm SURF gives the best results. This hybrid system works well because they play each other strengths and removes some of the drawbacks. The tracking point algorithm is good at detecting and compensating for small movements, but is prone to drift off target over longer periods. This can be rectified by the face recognition algorithm. The face recognition, on the other hand, is resource intensive and cannot be run on each frame in a real time system, but since the tracking point algorithm can compensate for small movement in the intervening time it does not need to run often.

This hybrid system works with reasonable accuracy, with an error rate, meaning a discarded recording, in an estimated 20 % of cases with limited movement and as low as 1-2 % in cases with no movement.

The tracking system has still a great potential of improvement in future works. It can not yet cope with excessive motion and has a too high error rate. We believe an effective tracking system is the biggest challenge if one is to create a functional non-contact system as of now.

Recording environment and video quality

We have tested the PPG HR detection system under different circumstances and it is found that video quality such as resolution and price of recording device does not affect results in any way. This leads the way in using cheap consumer grade products for developing a functional system. High compression formats are to be avoided, since they remove necessary color information from the frames. Medium compressed formats such as wmv is a reasonable compromise between compression and file size that does not remove too much color information.

The surroundings are found to affect the results the most. A stable ambient light source is needed and a calm recording environment without excessive disturbance. The largest source of erroneous detections has been proved to be due to the motion of the subject during recording.

References

[1] Wikimedia commons. Web. commons.wikimedia.org.

[2] The british mediacl association illustrated medical dictionary, 2nd ed, 2007.

[3] J Ando, A Kawarada, M Shibata, K Yamakoshi, and A Kamiya. Pressure-volume relation-ships of finger arteries in healthy subjects and patients with coronary atherosclerosis measured non-invasively by photoelectric plethysmography. *Japanese Circulation Journal*, 55(6):567–575, 1991.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359, 2008.

[5] J. F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11:157–192, 1999.

[6] J. F. Cardoso. Jade in matlab. Online, May 2005. Implementation of ICA algorithm (JADE).

[7] Pech Y S Cheang and Peter R Smith. An overview of non-contact photoplethysmography. *Electronic Systems And Control Division Research*, 1:57–59, 2003.

[8] W. Chen. Lossy to lossless compression of ecg by using integer wavelet transform. *Medical and Biological Engineering and Computing*, 38:569–574, 2000.

[9] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[10] Franklin Crow. Summed-area tables for texture mapping. *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 207–212, 1984.

[11] Maureen C.Stone. A survey of color for computer graphics. Course SIGGRAPH, August 2001.

[12] W. Cui, L.E. Ostrander, and B.Y. Lee. In vivo reflectance of blood and tissue as a function of light wavelength. *IEEE Transactions on Biomedical Engineering*, 37(6):632–639, June 1990.

[13] Mark D. Fairchild. *Color Appearance Models*. John Wiley & Sons, (second edition) edition, 2005. ISBN 0470012692.

[14] A Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.

[15] M. Hirai, S.L. Nielsen, and N.A. Lassen. Blood pressure measurements of all five fingers by strain gauge plethysmography. *Scandinavian Joural of Clinical & Laboratory Investigation*, 36(7):627–632, 1976.

[16] Ievgen Khvedchenia. Comparison of the opencv's features detection algorithms. Web, January 2011. http://computer-vision-talks.com/2011/01/comparison-of-the-opencvs-feature-detection-algorithms-2/.

[17] D.J. Korteweg. Uber die fortpflanzungsgeschwindigkeit des schalles in elastischen rohren. *Ann. Phys. Chem*, 5:527–537, 1878.

[18] B M McCarthy, B O'Flynn, and A Mathewson. An investigation of pulse transit time as a non-invasive blood pressure measurement method. *Journal of Physics: Conference Series*, 307:1–6, 2011.

[19] Adr. Isebree Moens. *Over de voortplantingssnelheid van den pols [On the speed of propagation of the pulse]*. PhD thesis, Leiden, 1877.

[20] U.S. Department of Health and Human Services. Pulse. web, Jan 2011.

[21] Folkehelsteinstituttet (Norwegian Institute of Public Health). Cardiovascular disease mortality in norway - fact sheet. Website, November 2012.

[22] Open CV. *OpenCV Documentation*, 2012.

[23] Ming-Zher Poh, Daniel J. McDuff, and Rosalind W. Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics Express*, 18:10762–10774, 2010.

[24] Andrews Reisner, Phillip A. Shaltis, Devin McCombie, and H. Harry Asada. Utility of the photoplethysmogram in circulatory monitoring. *Anesthesiology*, 108(5):950–958, May 2008.

[25] Robert A. Robergs and Roberto Landwehr. The surprising history of the "hrmax=220-age" equation". *Journal of Exercise Physiology 5 (2)*, 5:9, 2002.

[26] Walter N. Durán Ronald F. Tuma and Klaus Ley. *Microcirculation*. Elsevier Inc, (second edition) edition, 2008. ISBN: 978-0-12-374530-9.

[27] Robin P Smithb, Jérôme Argoda, Jean-Louis Pépina, and Patrick A Lévya. Pulse transit time: an appraisal of potential clinical applications. *Thorax An International Journal Of Respiratory Medicine*, 54:452–457, 1999.

[28] Åsmund Rustand. Ambient-light photoplethysmography. Master's thesis, NTNU, 2012.

[29] Milan Sonka, Vaclav Hlavac, and Rodger Boyle. *Image Processing, Analysis, and Machine Vision*. CL Engineering, 2007.

[30] Tsouri, Kyal, Dianat, and Mestha. Constrained independent component analysis approach to nonobtrusive pulse rate measurements. *Journal of Biomedical Optics*, 17:1–4, 2012.

[31] National Health Services UK. What is blood pressure? Web, March 2012.

[32] VanPutte, Reagen, and Russo. *Seeley's Essentials of Anatomy & Physiology*. McGraw-Hill, (7 edition) edition, 2010. ISBN 978-0-07-122006-4.

[33] Wim Verkruysse, Lars O. Svaasand, and J. Stuart Nelson. Remote plethysmography imaging using ambient light. *Optics Express*, 16:21434–21445, 2008.

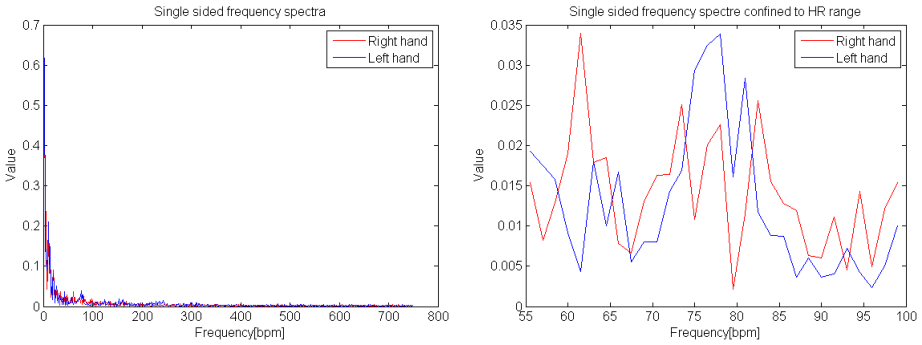
[34] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 1:511–518, 2001.

[35] W. G. Zijlstra, A. Buursma, and W.P. Meeuwsen van der Roest. Absorption spectra of human fetal and adult oxyhemoglobin, de-oxyhemoglobin, carboxyhemoglobin, and methemoglobin ,. *Clinical Chemistry*, 37(9):1633–1638, September 1991.

A Additional test data

A.1 Experiment 1

In this section is the second part of experiment 1. FFT can be observed in figure A.1 and trace of the Green RBG channel can be observed in figure A.1.



(a) Frequency spectra

(b) Frequency spectra confined to HR range

Figure A.1 – Frequency spectra of experiment 1. Benchmark test

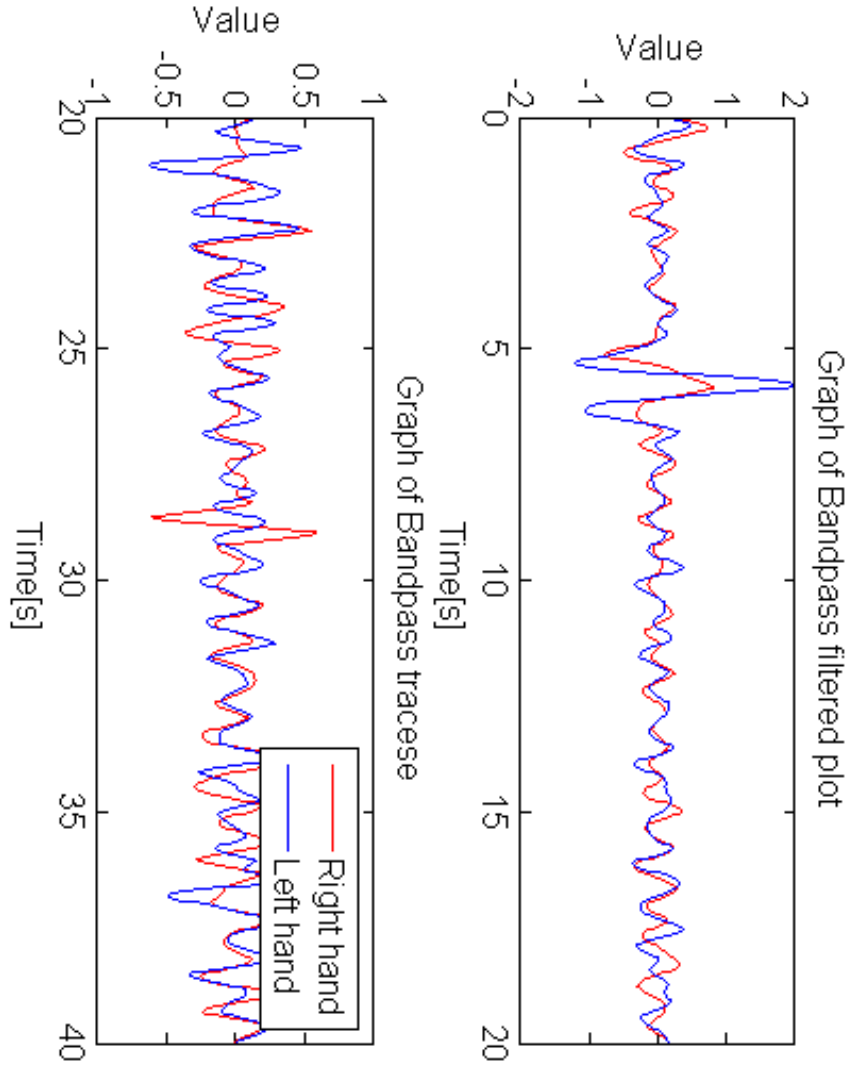


Figure A.2 – Experiment 1, benchmark test. Trace of Green channel, bandpass filtered in HR range

A.2 Experiment 2

This is the second attempt at experiment 2. In this experiment the removal of occlusion happens between part 2 and 3. FFT of part one can be observed in figure A.3, part two in A.4 and part three in A.5. A bandpassed trace of the Green channel can be observed in figure A.6 to A.8.

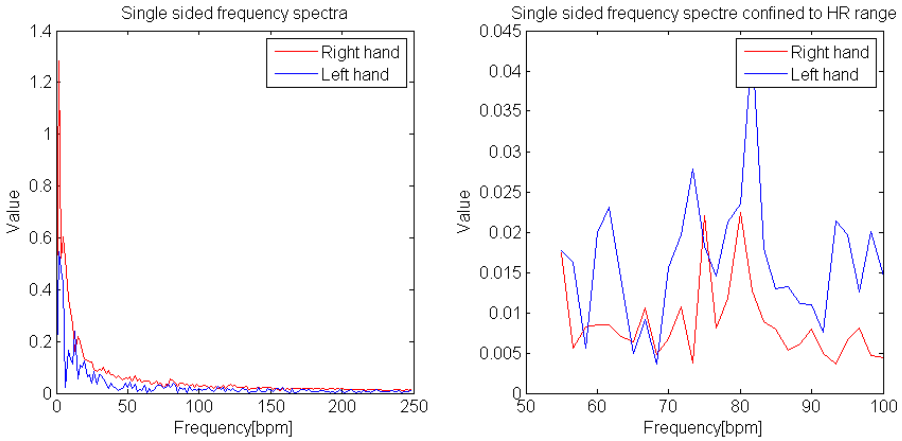


Figure A.3 – Frequency spectra of experiment 2, part 1 of 3. Right hand fully occluded.

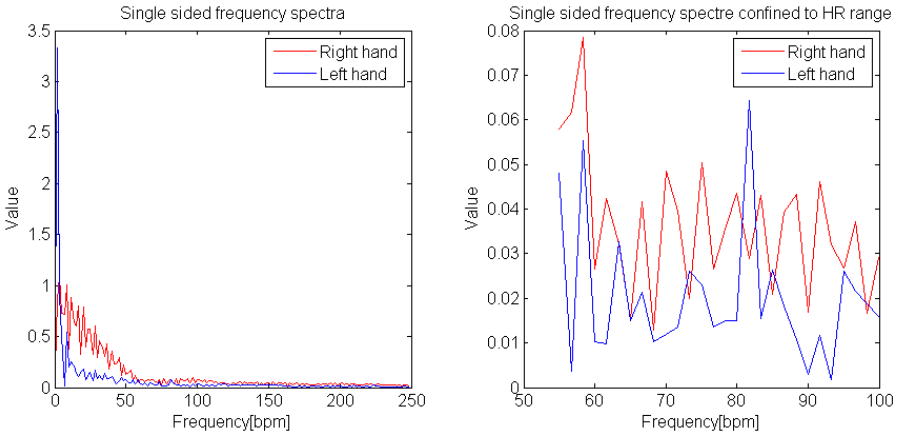


Figure A.4 – Frequency spectra of experiment 2, part 2 of 3. Right hand fully occluded.

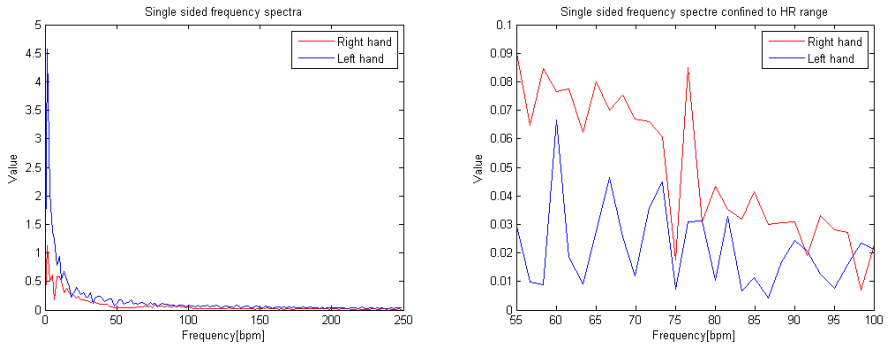


Figure A.5 – Frequency spectra of experiment 2, part 3 of 3. Occlusion is removed

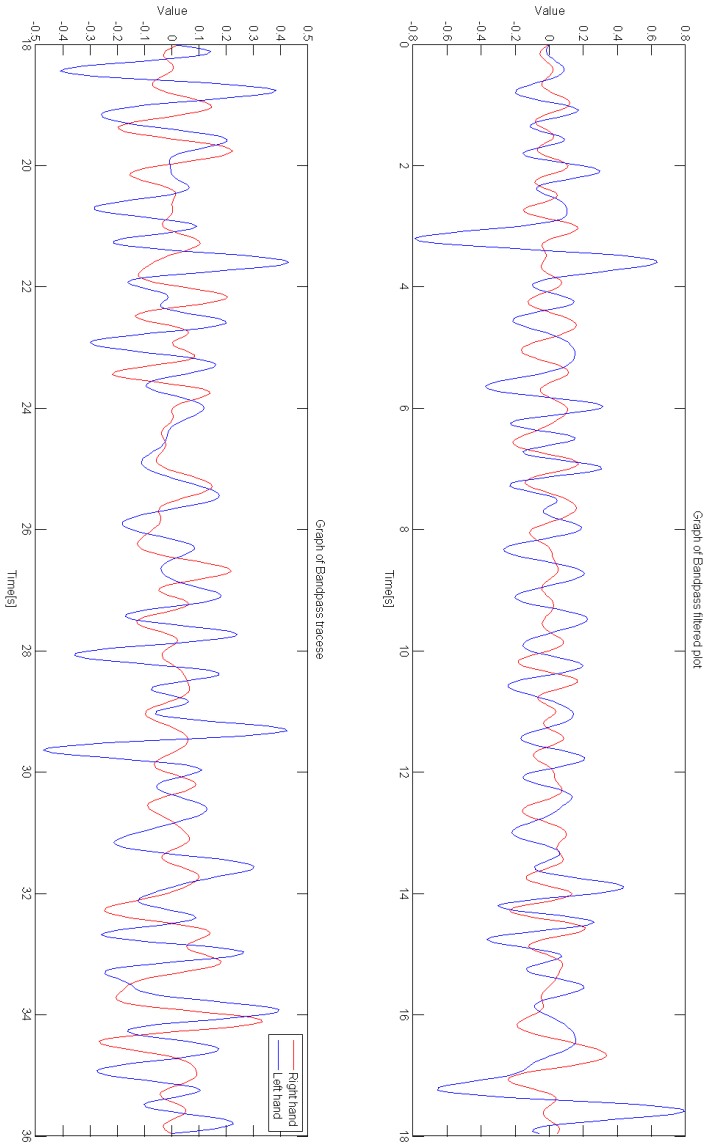


Figure A.6 – Trace of experiment 2, part 1 of 3. Trace of Green channel, bandpass filtered in HR range

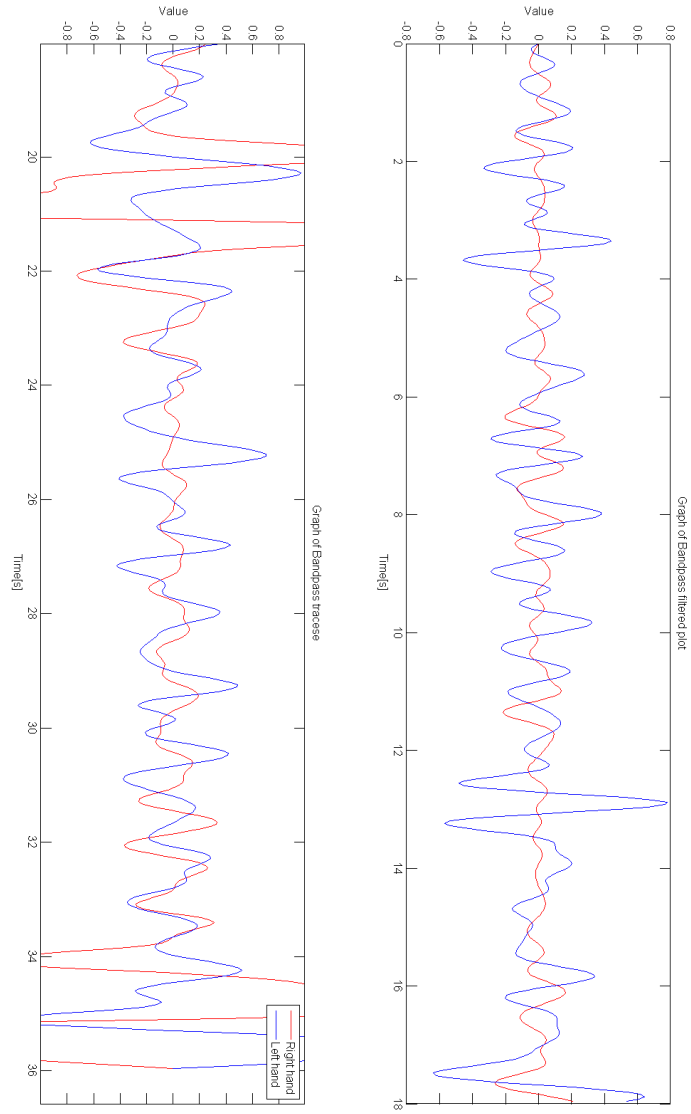


Figure A.7 – Trace of experiment 2, part 2 of 3. Right hand fully occluded. Trace of Green channel, bandpass filtered in HR range

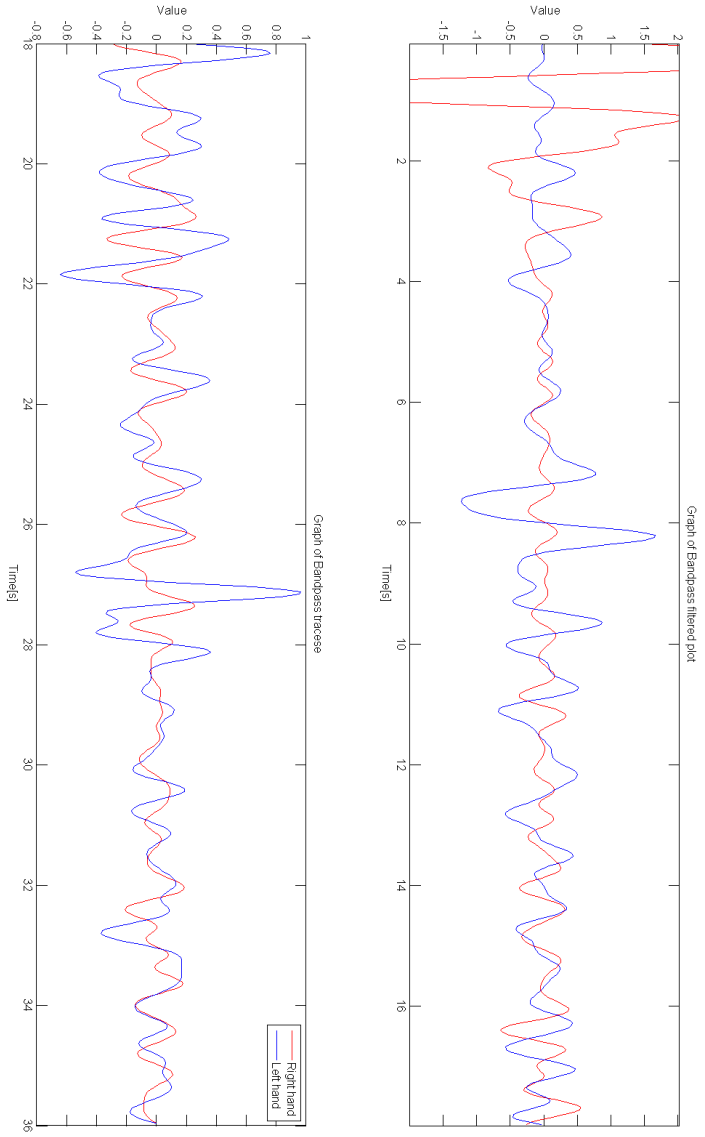


Figure A.8 – Trace of experiment 2, part 3 of 3. Occlusion is removed. Trace of Green channel, bandpass filtered in HR range

A.3 Experiment 3

This is the second attempt at experiment 3. In part 1 the cuff is not inflated to get benchmark results. In part 2 the cuff is gradually inflated and then released over a period of approx. 30 seconds. In part 3 the cuff is fully deflated and we get the unoccluded results again.

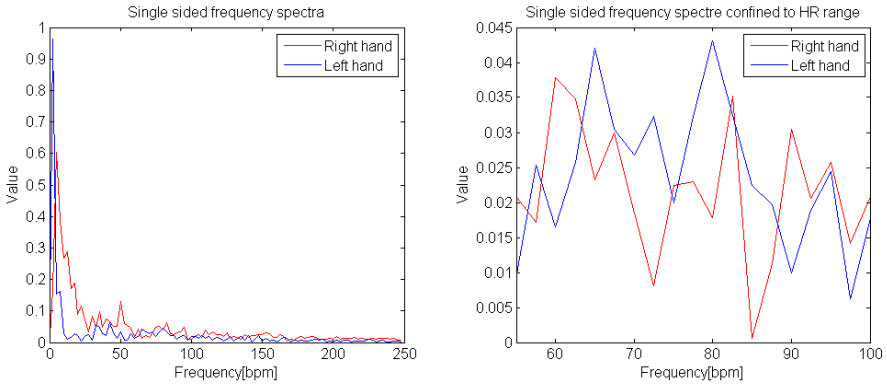


Figure A.9 – Frequency spectra of experiment 3, part 1 of 3

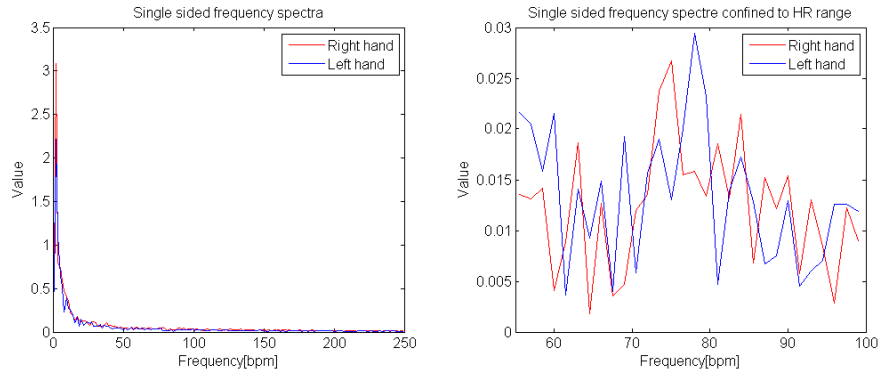


Figure A.10 – Frequency spectra of experiment 3, part 2 of 3

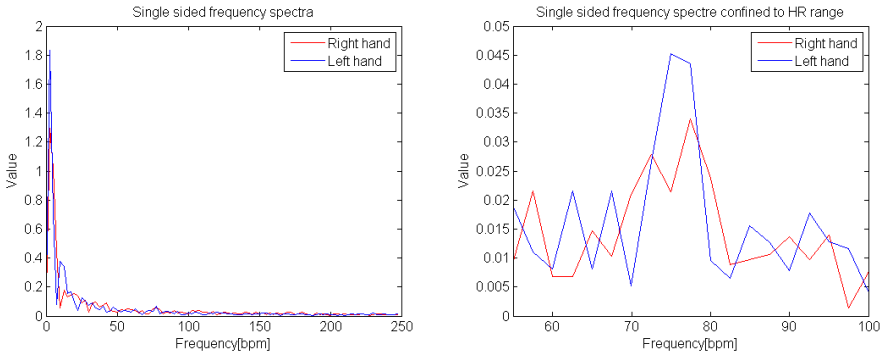


Figure A.11 – Frequency spectra of experiment 3, part 3 of 3

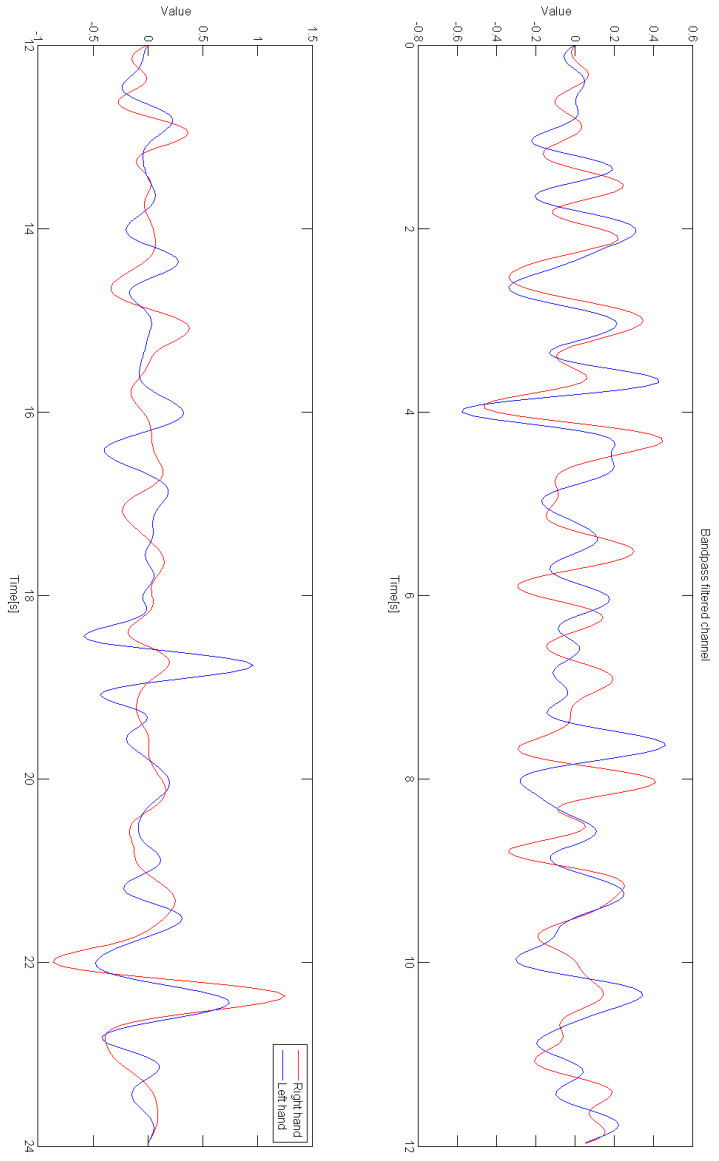


Figure A.12 – Trace of experiment 3, part 1 of 3. Cuff is not inflated and there is no occlusion. Trace of Green channel, bandpass filtered in HR range

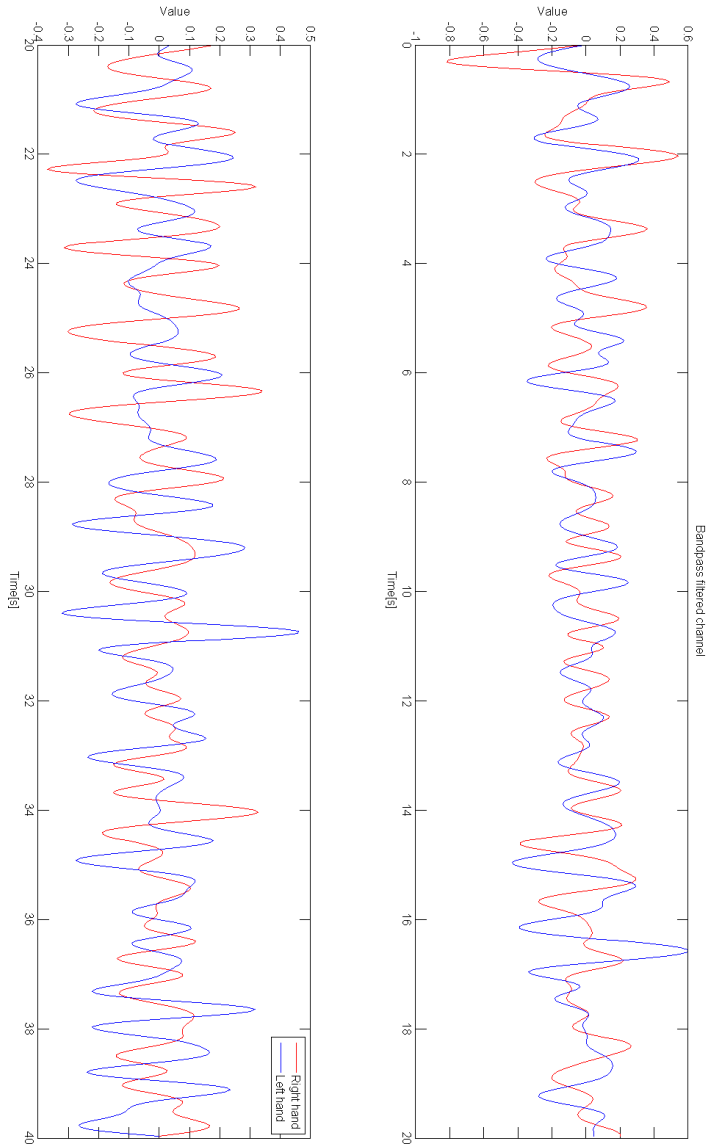


Figure A.13 – Trace of experiment 3, part 2 of 3. The medical cuff is gradually inflated over a period of 20 seconds and is deflated after 30. Trace of Green channel, bandpass filtered in HR range

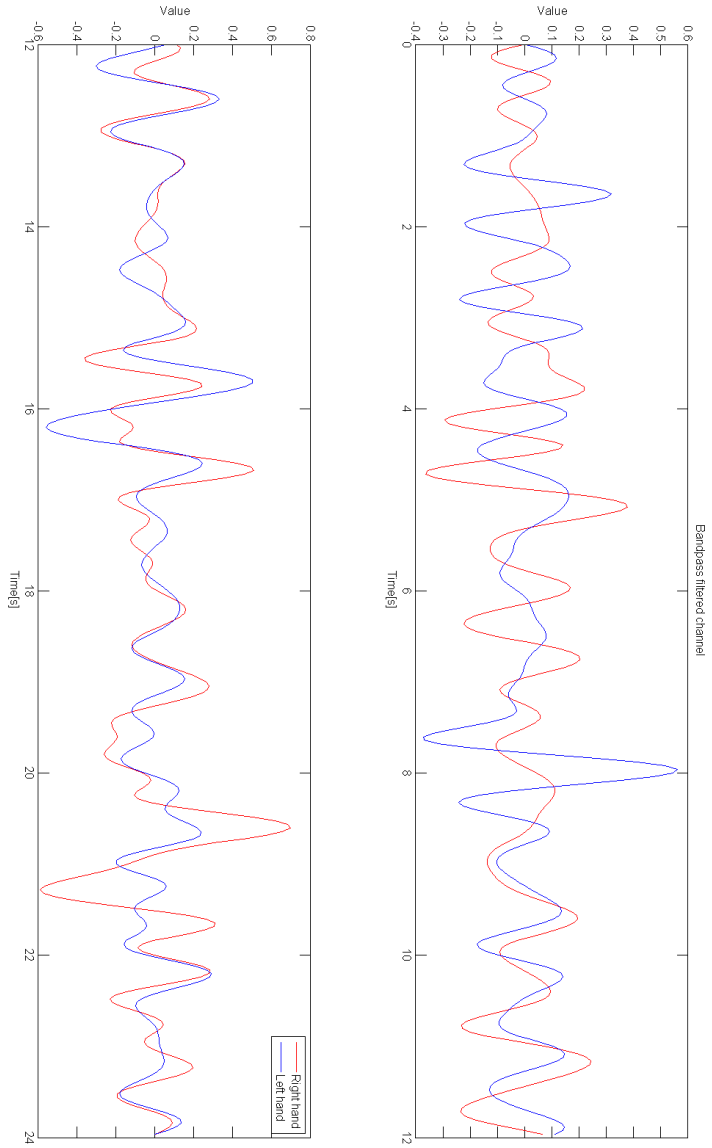


Figure A.14 – Trace of experiment 3, part 3 of 3. Cuff is deflated and there is no occlusion. Trace of Green channel, bandpass filtered in HR range

B C++ Code samples

B.1 Main function

B.1.1 Header file

```
/*
 * main.h
 *
 *  Created on: 11. sep. 2012
 *      Author: Nikolai
 */

#ifndef MAIN_H_
#define MAIN_H_

//Standar includes
#include <iostream>
#include <cstdio>
#include <string>



//DSP includes
//#include "Dsp.h"

//Open CV includes
#include "opencv2\highgui\highgui.hpp"
#include "opencv2\core\core.hpp"
#include "opencv2\opencv.hpp"
#include "opencv2\photo\photo.hpp"


//Project spesific includes
#include "functions_facemanipulations.h"
#include "import_video.h"


//Namespaces
using namespace std;
using namespace cv;

//Function decleration

void dispWebCam(CvCapture* capture);
void dispCamStat(CvCapture *capture);
void dispWebCamWithFaceDet(CvCapture* capture);
bool reactOnButtonPush(int key,CvCapture*capture);
void detect_and_draw( IplImage* image );// Function prototype for
    detecting and drawing an object from an image
void detectAndDisplay(Mat frame);



#endif /* MAIN_H_ */
```

Listing 1 – Header file for main function. Includes the nessecary libraries

B.1.2 Source file

```
#include "main.h"

//Global variables
const String face_cascade_name ="haarcascades\\
    haarcascade_frontalface_alt.xml";
const String eyes_cascade_name="haarcascades\\
    haarcascade_eye_tree_eyeglasses.xml";
const String face_side_cascade_name="haarcascades\\
    haarcascade_profileface.xml";
CascadeClassifier face_cascade;
CascadeClassifier eyes_cascade;
CascadeClassifier face_side_cascade;

int main()
{
int inputChoice=-1;
int detectionmode=-1;
int numberOfROI=-1;
int inputVar=-1;
bool usingMotionCompInput=false;
VideoCapture video;

cout<<"1. Webcam"<<endl<<"2. File"<<endl;
//cin>>inputChoice;
inputChoice=1;//cheat mode

cout<<"Detection mode?"<<endl;
cout<<"1. Automatic "<<endl<<"2. Manual"<<endl;
//cin>>detectionmode;
detectionmode=1;//cheat mode

cout<<"Number of ROI"<<endl;
//cin>>numberOfROI;
numberOfROI=1;//cheat mode

if(numberOfROI<1 || numberOfROI>10)
{
  cout<<"Erronous number of ROI inputted, 1 ROI selected for you"<<endl
      ;
  numberOfROI=1;
}

cout<<"Using motioncomp?"<<endl;
cout<<"1. Yes"<<endl;
cout<<"2. No"<<endl;
//cin>>inputVar;

if(inputVar==1)
{
  usingMotionCompInput=true;
}
else
{
  usingMotionCompInput=false;
}
usingMotionCompInput=true;//cheat mode
if(inputChoice >2 || inputChoice<1)
{
  cout<<"You are to retarded to choose, webcam decided"<<endl;
  inputChoice=1;
}
```

```
if(inputChoice==1)
{
  video=setupcamera(video);
  if(video.isOpened()==0)
  {
    cout<<"Not able to detect or setup webcam"<<endl;
    return 0;
  }
  if(detectionmode==1)
  {
    automaticROIselectWebCam(video,false);
  }
  else if(detectionmode==2)
  {
    manualROIselectWebCam(video,false);
  }
  else
  {
    cout<<"Error"<<endl;
  }
}
else
{
  string videofile_name="d";
  string crap;

  cout<<"Name of video file:"<<endl;
  getline(cin,crap);//catch useless newline
  getline(cin,videofile_name);
  video.open(videofile_name);
  if(!video.isOpened())
  {
    cout<<"File not found, using default"<<endl;
    videofile_name="C:\\Users\\Nikolai\\Videos\\Master\\wmv\\subject2.
        avi";
    video.open(videofile_name);
    if(!video.isOpened())
    {
      cout<<"Default video file not found, terminating"<<endl;
      waitKey(0);
      system("pause");
      return 0;
    }
    if(detectionmode==1)
    {
      dispWebCamWithFaceDet(video);
    }
    else if(detectionmode==2)
    {
      //automaticROIselectWebCam(video,true);
      manualROIselectWebCam(video,true,numberOfROI,usingMotionCompInput
          );
    }
    else
    {
      cout<<"Well this was not supposed to happen, this is akward..."<<
          endl;
    }
  }
}
waitKey(0);
video.~VideoCapture();
```

```
return 0;
}
```

Listing 2 – Main function. Used for importing video from either file or webcam

B.2 Placing ROI and tracking algorithm

B.2.1 Setup webcam

```
VideoCapture setupcamera(VideoCapture video)
{

  //init camera
  video.open(0);

  if(!video.isOpened())
  {
    cout<<"Not able to init webcam"<<endl;
    return 0;
  }
  cout<<"Frame FPS= "<<video.set(CV_CAP_PROP_FPS, 30)<<endl;
  cout<<"Frame Mode= "<<video.set(CV_CAP_PROP_MODE, 0)<<endl;

  //cout<<"Frame height= "<<video.set(CV_CAP_PROP_FRAME_HEIGHT, 360)<<
      endl;
  //cout<<"Frame width= "<<video.set(CV_CAP_PROP_FRAME_WIDTH, 640)<<
      endl;


  //cout<<"Frame height= "<<video.set(CV_CAP_PROP_FRAME_HEIGHT, 720)<<
      endl;
  //cout<<"Frame width= "<<video.set(CV_CAP_PROP_FRAME_WIDTH, 1280)<<
      endl;

  video.set(CV_CAP_PROP_BRIGHTNESS,77);
  video.set(CV_CAP_PROP_WHITE_BALANCE_BLUE_U,3950);
  video.set(CV_CAP_PROP_SATURATION,40);
  video.set(CV_CAP_PROP_EXPOSURE,-8);
  video.set(CV_CAP_PROP_CONTRAST,6);
  video.set(CV_CAP_PROP_FOCUS,17);
  //int=CV_FOURCC('D','I','B',' ');
  video.set(CV_CAP_PROP_FOURCC,CV_FOURCC('D','I','B',' '));

  return video;
}
```

Listing 3 – Funtion for setting up webcam

B.2.2 Display webcam settings

```
void dispCamStat(VideoCapture Video)
{
  cout<<"Width= "<<Video.get(CV_CAP_PROP_FRAME_WIDTH)<<endl;
  cout<<"Height= "<<Video.get(CV_CAP_PROP_FRAME_WIDTH)<<endl;
  cout<<"FPS= "<<Video.get(CV_CAP_PROP_FPS)<<endl;
```

```
  int ex = static_cast<int>(Video.get(CV_CAP_PROP_FOURCC));     // Get
      Codec Type- Int form
  char EXT[] = {ex & 0XFF , (ex & 0XFF00) >> 8,(ex & 0XFF0000) >> 16,(
      ex & 0XFF000000) >> 24, 0}; // Transform from int to char via
      Bitwise operators

  cout<<"Codec= "<<EXT<<endl;
  cout<<"Timecode= "<<Video.get(CV_CAP_PROP_FORMAT)<<endl;
  cout<<"Brightness= "<<Video.get(CV_CAP_PROP_BRIGHTNESS)<<endl;
  cout<<"Contrast= "<<Video.get(CV_CAP_PROP_CONTRAST)<<endl;
  cout<<"Saturation= "<<Video.get(CV_CAP_PROP_SATURATION)<<endl;
  cout<<"Hue= "<<Video.get(CV_CAP_PROP_HUE)<<endl;
  cout<<"Gain= "<<Video.get(CV_CAP_PROP_GAIN)<<endl;
  cout<<"Convert RGB= "<<Video.get(CV_CAP_PROP_CONVERT_RGB)<<endl;
  cout<<"Mode= "<<Video.get(CV_CAP_PROP_MODE)<<endl;

}
```

Listing 4 – Function for displaying webcam state

B.2.3 Placing ROI manually

```
bool ROIrectSelect(VideoCapture video, Mat frame, Rect* ROIrect, bool
    fromFile)
{
  int key=0;
  bool ROIselected=false;
  Mat framecopy;
  int moveSpeed=4;
  while(key !='q')
  {
    if(!fromFile)
    {
      video>>frame;
    }
    frame.copyTo(framecopy);
    ROIrect->height=global_ROItrackBar*(frame.rows/40);
    ROIrect->width=global_ROItrackBar*(frame.cols/40);
    rectangle(framecopy,*ROIrect,CV_RGB(0,0,255),2,5,0);
    imshow("Video",framecopy);
    key=waitKey(10);

    if (key==2424832)//left key
    {
      if(ROIrect->x>0)
      {
        ROIrect->x-= moveSpeed;
      }
    }
    if (key==2490368)//up key
    {
      if(ROIrect->y>0)
      {
        ROIrect->y-= moveSpeed;
      }
    }

    if (key==2555904)//right key
    {
      if(ROIrect->x+ROIrect->width<frame.cols-1)
```

```
      {
        ROIrect->x+= moveSpeed;
      }
    }
    if (key==2621440)//down key
    {
      if(ROIrect->y+ROIrect->height<frame.rows-1)
      {
        ROIrect->y+= moveSpeed;
      }
    }
    if (key==32)//spacebar
    {

      ROIselected=true;
      break;
    }
    if(key=='q')
    {
      global_quit=true;
    }
  }
  return ROIselected;
}
```

Listing 5 – Function for placing the ROI manually using keypad

B.2.4 Automatic placement of single ROI using face recognition

```
void automaticROIselectWebCam(VideoCapture video, bool fromFile)
{

  int key=0;
  int nFacesDetected=0;
  int numberofROI=0;
  Mat frame;
  Mat frameCopyMat;
  Mat frameGray;

  vector<Rect> facesRectVector;
  SurfFeatureDetector surfdetector(400,4,2,0,0);//setting up SURF
      properties
  outputHR(3,true);//duping old data

  if(face_cascade.load(face_cascade_name)==0)//Detecting that cascades
      exist
  {
    cout<<"Error loading face cascade, exiting"<<endl;
    system("pause");
    return;
  }

  vector<Rect> ROIrectVector;

  bool ROIselected=false;
  bool usingMotionComp=true;//change this to use motioncompansation
  bool refreshPosUsingFaceRec=true;
  int faceRefreshIntervall=150;// how often you refresh faceposition

  Rect ROIrect;
```

```
int FPS=video.get(CV_CAP_PROP_FPS);
cout<<"Framerate= "<<FPS<<endl;
//FPS=60;//for special video

namedWindow("Video",CV_WINDOW_AUTOSIZE);
createTrackbar("Windowsize","Video",&global_windowsize,1000,
    onROITrackbarPushWindSize);
if (fromFile)
{
  int nFramesInVid=video.get(CV_CAP_PROP_FRAME_COUNT);
  createTrackbar("Video position","Video",&global_videoPosition,
      nFramesInVid,onVideoPositionPush);
}

for(int i=0;i<50;i++)//disregarding 50 first frames
{
  try
  {
    video>>frame;
  }
  catch(cv::Exception &e)
  {
    cout<<e.what()<<endl;
    return;
  }
  imshow("Video",frame);
  //waitKey(1/FPS);
}
while(key!='q' && !global_quit)
{
  key=waitKey(1);
  if(frame.empty())
  {
    break;
  }
  if(!ROIselected)//here we detect face
  {
    if(fromFile)//Saying number position and setting trackbar
        position
    {
      int videoPosition=video.get(CV_CAP_PROP_POS_FRAMES);
      if(videoPosition<0||videoPosition>video.get(
          CV_CAP_PROP_FRAME_COUNT))
      {
        cout<<"Error in reading videopos"<<endl;
      }
      else
      {
        global_videoPosition=videoPosition;
        cout<<"Video position is "<<videoPosition<<endl;
        setTrackbarPos("Video position","Video",global_videoPosition)
            ;

        cout<<"Push key to continue"<<endl;
        key=waitKey(0);
        if (key=='q')
        {
          cout<<"Exit pushed, exiting"<<endl;
          return;
        }
      }
```

```
    }
    bool isFacesDetected=false;//Here we detect faces and number of
        faces
    facesRectVector.clear();
    ROIrectVector.clear();
    while(!isFacesDetected)
    {
      detectingFaceFunc(video, frame, frameGray, facesRectVector,
          isFacesDetected, ROIselected, nFacesDetected, numberofROI,
          ROIrectVector, key);
    }
    cout<<"Select windowsize now"<<endl;
    key=waitKey(0);
  }

  if(fromFile)
  {
    video.set(CV_CAP_PROP_POS_FRAMES,global_videoPosition);
    int nFramesInVid=video.get(CV_CAP_PROP_FRAME_COUNT);
    if(nFramesInVid>0)//Checking if windowsize is to large for file
    {
      if(global_windowsize>(nFramesInVid-global_videoPosition))
      {
        cout<<"global_windowsize to large, using length of video"<<
            endl;
        global_windowsize=nFramesInVid-global_videoPosition;
        cout<<"global_windowsize now= "<<global_windowsize<<endl;
      }
    }
    else
    {
      cout<<"Error in reading number of frames"<<endl;
    }
  }
  vector< vector<Mat> > ROIvector(numberofROI,vector<Mat>(
      global_windowsize));//declaring and initializing it here for
      changing number of windowsize
  if(ROIselected)
  {
    if(usingMotionComp==false)//if not using motioncomp, just stores
        windowsize number of frames
    {
      for(int i=0;i<global_windowsize;i++)
      {
        video>>frame;
        if(frame.empty())
        {
          break;
        }
        for(int j=0;j<numberofROI;j++)
        {
          Mat temp;
          frame(ROIrectVector[j]).copyTo(temp);
          ROIvector[j][i]=temp;
          rectangle(frame,ROIrectVector[j],CV_RGB(255,0,0),2,5,0);
          rectangle(frame,facesRectVector[j],CV_RGB(0,0,255),2,3,0);
        }
        imshow("Video",frame);
        key=waitKey(1);
      }

    }
```

```
else//if using motioncomp
{
  Mat frame_gray;
  Mat orgFrame;
  vector<KeyPoint> features;
  vector<Mat> faces;
  vector<Point>ROIcordVector(numberofROI);
  vector<Point>prevPointVector(numberofROI);
  vector<bool> firstDetect(numberofROI);

  for(int p=0;p<numberofROI;p++)//initializer
  {
    ROIcordVector[p]=ROIrectVector[p].tl();
    prevPointVector[p]=Point(0,0);
    firstDetect[p]=true;
  }

  for(int i=0;i<global_windowsize;i++)//Main for loop
  {
    video>>frame;
    if(frame.empty())
    {
      break;
    }
    frame.copyTo(frameCopyMat);//used for drawing on
    for(int k=0;k<numberofROI;k++)
    {
      cvtColor(frame(ROIrectVector[k]),frame_gray,CV_BGR2GRAY);
      surfdetector.detect(frame_gray,features);
      for(unsigned int j=0;j<features.size();j++)//changing
          coordinates to match frame
      {
        Point temp=features[j].pt;
        Point temp2=temp+ROIcordVector[k];
        features[j].pt=temp2;
      }
      drawKeypoints(frameCopyMat,features,frameCopyMat,Scalar::
          all(255));
      if(!features.empty())
      {
        float max=features[0].response;
        int maxIndex=0;

        Point firstpoint=features[0].pt;
        Point difference=firstpoint-prevPointVector[k];
        float minMagnitude=sqrt(float(difference.x*difference.x+
            difference.y+difference.y));

        for(unsigned int j=1;j<features.size();j++)//traverse
            list
        {
          if(!firstDetect[k])//find closest and strongest
          {
            Point temp=features[j].pt;
            difference=temp-prevPointVector[k];
            float magnitude=sqrt(float(difference.x*difference.x+
                difference.y+difference.y));
            if(magnitude<minMagnitude)
            {
              minMagnitude=magnitude;
              maxIndex=j;
            }
```

```
        }
        else
        {
          if(features[j].response>max)//find strongest
          {
            maxIndex=j;
            max=features[j].response;
          }
        }
      }
      float maxMovement=8;
      float minMovement=1;

      circle(frameCopyMat,features[maxIndex].pt,4,Scalar
          (0,0,255),5,1);
      Point strongestTrackPoint=features[maxIndex].pt;//selects
           the point to follow
      if(!firstDetect[k])// && minMagnitude>maxMovement &&
          minMagnitude<minMovement);
      {
        if(minMagnitude>maxMovement)
        {
          cout<<"To large deviation"<<endl;
          strongestTrackPoint=prevPointVector[k];
        }
        else if(minMagnitude<minMovement)
        {
          cout<<"To little change"<<endl;
          strongestTrackPoint=prevPointVector[k];
        }
      }
      Point change=strongestTrackPoint-prevPointVector[k];
      if(!firstDetect[k])
      {
        if(!(ROIrectVector[k].x+change.x+ROIrectVector[k].width
             > frame.cols ||ROIrectVector[k].x+change.x<0) &&
            !(ROIrectVector[k].y+change.y+ROIrectVector[k].
            height > frame.rows||ROIrectVector[k].y+change.y<0)
            )//so new ROI dont exceed frame
        {
          ROIrectVector[k].x=ROIrectVector[k].x+change.x;
          ROIrectVector[k].y=ROIrectVector[k].y+change.y;
          ROIcordVector[k]=ROIrectVector[k].tl();
          prevPointVector[k]=strongestTrackPoint;
        }
        else
        {
          cout<<"New ROI out of bounds"<<endl;
        }
      }
      else
      {
        prevPointVector[k]=strongestTrackPoint;
        firstDetect[k]=false;
      }
    }
    Mat temp;
    frame(ROIrectVector[k]).copyTo(temp);
    ROIvector[k][i]=temp;
    rectangle(frameCopyMat,ROIrectVector[k],CV_RGB(255,0,0)
        ,2,5,0);
  }
```

```
          if (refreshPosUsingFaceRec && i%faceRefreshIntervall==0)
          {
            int nROIbeforeFacedetect=numberofROI;
            bool isFacesDetected=false;
            ROIrectVector.clear();
            facesRectVector.clear();
            while(!isFacesDetected)
            {
              detectingFaceFunc(video,frame,frameGray,facesRectVector,
                  isFacesDetected,ROIselected,nFacesDetected,
                  numberofROI,ROIrectVector,key);
            }
            if(nROIbeforeFacedetect < numberofROI)//ignore if more
                faces appear after first detect
            {
              numberofROI=nROIbeforeFacedetect;
            }
            for (int t=0;t<firstDetect.size();t++)
            {
              firstDetect[t]=true;
            }
          }

          imshow("Video",frameCopyMat);
          key=waitKey(1);
          if (key=='q')
          {
            cout<<"Quiting"<<endl;
            global_quit=true;
            return;
            break;
          }
          features.clear();
        }
      }
      ROIselected=true;//set false to select new ROI each run
      for(int i=0;i<numberofROI;i++)
      {
        if(!ROIvector[i].empty())
        {
          //viewDiffColorspace(windowsize,ROIvector[i],ROIvector[0].
              cols,ROIvector[0].rows,FPS);//if send to swapfile
          HRfromDiffColorspace(ROIvector[i].size(),ROIvector[i],FPS);//
              if send to HR detect SW
        }
        else
        {
          cout<<"End of file"<<endl;
          break;
        }
      }
      ROIvector.clear();
    }
  }
}
```

Listing 6 – Placing a single ROI over forehead of subject

B.2.5 Manually placing multiple ROI

,

```
void manualROIselectWebCam(VideoCapture video, bool fromFile,int nROI,
    bool motioncomp)
{
  int numberofROI=nROI;
  int key=0;
  bool OdaIsHot=true;
  Mat frame;
  Mat frameCopyMat;
  SurfFeatureDetector surfdetector(400,4,2,0,0);
  face_cascade.load(face_cascade_name);

  vector<Rect> ROIrectVector;

  bool ROIselected=false;
  bool usingMotionComp=motioncomp;//change this to use
      motioncompansation
  Rect ROIrect;
  int FPS=video.get(CV_CAP_PROP_FPS);
  cout<<"Framerate= "<<FPS<<endl;//TODO automatisert
  //FPS=60;//for special video

  namedWindow("Video",CV_WINDOW_AUTOSIZE);
  for(int i=0;i<50;i++)//disregarding 50 first frames
  {
    try
    {
      video>>frame;
    }
    catch(cv::Exception &e)
    {
      cout<<e.what()<<endl;
      return;
    }
    imshow("Video",frame);
  }

  ROIrect.x=frame.cols/2;//Setting manual ROI pos, initiate
  ROIrect.y=frame.rows/4;

  createTrackbar("ROI size","Video",&global_ROItrackBar,20,
      onROITrackbarPush);
  createTrackbar("Windowsize","Video",&global_windowsize,1000,
      onROITrackbarPushWindSize);
  if (fromFile)
  {
    int nFramesInVid=video.get(CV_CAP_PROP_FRAME_COUNT);
    createTrackbar("Video position","Video",&global_videoPosition,
        nFramesInVid,onVideoPositionPush);
  }
  while(key!='q' && !global_quit)
  {
    key=waitKey(10);
    if(frame.empty())
    {
      break;
    }
    if(!ROIselected)//here we select ROI
    {
      ROIrectVector.clear();

      if(fromFile)
      {
```

```
      int videoPosition=video.get(CV_CAP_PROP_POS_FRAMES);
      if(videoPosition<0||videoPosition>video.get(
          CV_CAP_PROP_FRAME_COUNT))
      {
        cout<<"Error in reading videopos"<<endl;
      }
      else
      {
        global_videoPosition=videoPosition;
        cout<<"Video position is "<<videoPosition<<endl;
        setTrackbarPos("Video position","Video",global_videoPosition)
            ;
      }
    }
    for(int i=0;i<numberofROI;i++)
    {
      ROIselected=ROIrectSelect(video, frame, &ROIrect,fromFile);//
          stay in this function until selected ROI
      ROIrectVector.push_back(ROIrect);
    }
  }

  if(fromFile)
  {
    video.set(CV_CAP_PROP_POS_FRAMES,global_videoPosition);
    int nFramesInVid=video.get(CV_CAP_PROP_FRAME_COUNT);
    if(nFramesInVid>0)//sjekker om vindusstørrelsen er for stor
    {
      if(global_windowsize>(nFramesInVid-global_videoPosition))
      {
        cout<<"global_windowsize to large, using length of video"<<
            endl;
        global_windowsize=nFramesInVid-global_videoPosition;
        cout<<"global_windowsize now= "<<global_windowsize<<endl;
      }
    }
    else
    {
      cout<<"Error in reading number of frames"<<endl;
    }
  }

  vector< vector<Mat> > ROIvector(numberofROI,vector<Mat>(
      global_windowsize));//declaring and initializing it here for
      changing number of windowsize
  double prevtimestamp=0;
  double prevtimestamptime=clock();
  if(ROIselected)
  {
    if(usingMotionComp==false)//if not using motioncomp, just stores
        windowsize number of frames
    {
      for(int i=0;i<global_windowsize;i++)
      {
        video>>frame;
        if(frame.empty())
        {
          break;
        }
        for(int j=0;j<numberofROI;j++)
        {
          Mat temp;
```

```
      frame(ROIrectVector[j]).copyTo(temp);
      ROIvector[j][i]=temp;
      rectangle(frame,ROIrectVector[j],CV_RGB(255,0,0),2,5,0);
    }
    imshow("Video",frame);
    key=waitKey(1);
  }
}
else//using motioncomp
{
  Mat frame_gray;
  Mat orgFrame;
  vector<KeyPoint> features;
  vector<Mat> faces;
  vector<Point>ROIcordVector(numberofROI);
  vector<Point>prevPointVector(numberofROI);
  vector<bool> firstDetect(numberofROI);

  for(int k=0;k<numberofROI;k++)//initializer
  {
    ROIcordVector[k]=ROIrectVector[k].tl();
    prevPointVector[k]=Point(0,0);
    firstDetect[k]=true;
  }
  for(int i=0;i<global_windowsize;i++)
  {
    video>>frame;
    if(frame.empty())
    {
      break;
    }
    frame.copyTo(frameCopyMat);//used for drawing on
    for(int k=0;k<numberofROI;k++)
    {
      cvtColor(frame(ROIrectVector[k]),frame_gray,CV_BGR2GRAY);
      surfdetector.detect(frame_gray,features);

      for(unsigned int j=0;j<features.size();j++)//changing
          coordinates to match frame
      {
        Point temp=features[j].pt;
        Point temp2=temp+ROIcordVector[k];
        features[j].pt=temp2;
      }
      drawKeypoints(frameCopyMat,features,frameCopyMat,Scalar::
          all(255));
      if(!features.empty())
      {
        float max=features[0].response;
        int maxIndex=0;

        Point firstpoint=features[0].pt;
        Point difference=firstpoint-prevPointVector[k];
        float minMagnitude=sqrt(float(difference.x*difference.x+
            difference.y+difference.y));

        for(unsigned int j=1;j<features.size();j++)//traverse
            list
        {
          if(!firstDetect[k])//find closest and strongest
          {
            Point temp=features[j].pt;
```

```
      difference=temp-prevPointVector[k];
      float magnitude=sqrt(float(difference.x*difference.x+
          difference.y+difference.y));
      if(magnitude<minMagnitude)
      {
        minMagnitude=magnitude;
        maxIndex=j;
      }
    }
    else
    {
      if(features[j].response>max)//find strongest
      {
        maxIndex=j;
        max=features[j].response;
      }
    }
  }
  float maxMovement=8;
  float minMovement=1;

  circle(frameCopyMat,features[maxIndex].pt,4,Scalar
      (0,0,255),5,1);
  Point strongestTrackPoint=features[maxIndex].pt;//selects
       the point to follow
  if(!firstDetect[k])// && minMagnitude>maxMovement &&
      minMagnitude<minMovement);
  {
    if(minMagnitude>maxMovement)
    {
      cout<<"To large deviation"<<endl;
      strongestTrackPoint=prevPointVector[k];
    }
    else if(minMagnitude<minMovement)
    {
      cout<<"To little change"<<endl;
      strongestTrackPoint=prevPointVector[k];
    }
  }
  Point change=strongestTrackPoint-prevPointVector[k];
  if(!firstDetect[k])
  {
    if(!(ROIrectVector[k].x+change.x+ROIrectVector[k].width
         > frame.cols ||ROIrectVector[k].x+change.x<0) &&
        !(ROIrectVector[k].y+change.y+ROIrectVector[k].
        height > frame.rows||ROIrectVector[k].y+change.y<0)
        )//so new ROI dont exceed frame
    {
      ROIrectVector[k].x=ROIrectVector[k].x+change.x;
      ROIrectVector[k].y=ROIrectVector[k].y+change.y;
      ROIcordVector[k]=ROIrectVector[k].tl();
      prevPointVector[k]=strongestTrackPoint;
    }
    else
    {
      cout<<"New ROI out of bounds"<<endl;
    }
  }
  else
  {
    prevPointVector[k]=strongestTrackPoint;
    firstDetect[k]=false;
```

```
            }
          }
          Mat temp;
          frame(ROIrectVector[k]).copyTo(temp);
          ROIvector[k][i]=temp;
          rectangle(frameCopyMat,ROIrectVector[k],CV_RGB(255,0,0)
              ,2,5,0);
        }
        imshow("Video",frameCopyMat);
        key=waitKey(1);
        features.clear();
      }
    }
    ROIselected=false;//to select new ROI
    for(int i=0;i<numberofROI;i++)
    {
      if(!ROIvector[i].empty())
      {
        //viewDiffColorspace(windowsize,ROIvector[i],ROIvector[0].
            cols,ROIvector[0].rows,FPS);//if send to swapfile
        HRfromDiffColorspace(ROIvector[i].size(),ROIvector[i],FPS);//
            if send to
      }
      else
      {
        cout<<"End of file"<<endl;
        break;
      }
    }
    ROIvector.clear();
  }
 }
}
```

Listing 7 – Function wery similar to automatic placement, enables one to place unlimited number of ROIs manually

B.2.6 Cascade classifier function for face detection

```
void detectingFaceFunc( VideoCapture video, Mat &frame, Mat frameGray,
    vector<Rect> &facesRectVector, bool &isFacesDetected, bool &
    ROIselected, int &nFacesDetected, int &numberofROI, vector<Rect> &
    ROIrectVector, int &key )
{
  video>>frame;
  imshow("Video",frame);
  cvtColor(frame,frameGray,CV_BGR2GRAY);
  equalizeHist(frameGray,frameGray);
  Size faceSizeDetectTresh=Size(frame.cols/7,frame.rows/6);
  face_cascade.detectMultiScale(frameGray,facesRectVector,1.1,2,0|
      CV_HAAR_SCALE_IMAGE,faceSizeDetectTresh);
  if (!facesRectVector.empty())
  {
    isFacesDetected=true;
    ROIselected=true;
    nFacesDetected=facesRectVector.size();
    numberofROI=nFacesDetected;
    for (unsigned int i=0;i<nFacesDetected;i++)
    {
      int faceWidth=facesRectVector[i].width;
```

```
      int faceHeight=facesRectVector[i].height;
      int faceXcord=facesRectVector[i].x;
      int faceYcord=facesRectVector[i].y;
      Rect tempROIrect;
      tempROIrect.x=faceXcord+floor(faceWidth*0.1)*2;
      if (tempROIrect.x<0)
      {
        tempROIrect.x=0;
      }
      tempROIrect.y=faceYcord-faceHeight*0.1;
      if (tempROIrect.y<0)
      {
        tempROIrect.y=0;
      }
      tempROIrect.width=floor(faceWidth*0.6);
      if (tempROIrect.width+tempROIrect.x>frame.cols)
      {
        tempROIrect.width=frame.cols-tempROIrect.x;
      }
      tempROIrect.height=floor(faceHeight*0.4);
      if (tempROIrect.height+tempROIrect.y>frame.rows)
      {
        tempROIrect.height=frame.rows-tempROIrect.y;
      }
      if (tempROIrect.width<0||tempROIrect.height<0)
      {
        cout<<"Error in cutting ROI of forehead"<<endl;
        return;
      }
      ROIrectVector.push_back(tempROIrect);
    }
  }
  else
  {
    cout<<"Face not detected"<<endl;
    key=waitKey(5);
    if (key=='q')
    {
      cout<<"Exiting"<<endl;
      return;
    }
  }
}
```

B.3 Automatic HR detection

B.3.1 Initial signal processing

```
,
void viewDiffColorspace( const int windowsize, vector<Mat> ROIvector,
    int nColums, int nRows,int FPS)//using this to view and dump data
    to other programs
{
  Mat HSVimage;
  Mat HSLimage;
  Mat YCrCbimage;
  Mat RGBimage;
  vector<Mat> RGBindividual;
  vector<Mat> HSVindividual;
  vector<Mat> HSLindividual;
```

```
vector<Mat> YCrCbindividual;
vector<vector<Mat> > RGBcomponentMatVector;
vector<vector<Mat> > HSVcomponentMatVector;
vector<vector<Mat> > HSLcomponentMatVector;
vector<vector<Mat> > YCrCbcomponentMatVector;
Scalar RGBcomponentScalar;
Scalar HSVcomponentScalar;
Scalar HSLcomponentScalar;
Scalar YCrCbcomponentScalar;
vector<vector<float> > RGBcomponentValueVector(3,vector<float>(
    windowsize));
vector<vector<float> > HSVcomponentValueVector(3,vector<float>(
    windowsize));
vector<vector<float> > HSLcomponentValueVector(3,vector<float>(
    windowsize));
vector<vector<float> > YCrCbcomponentValueVector(3,vector<float>(
    windowsize));
for(int i=0;i<windowsize;i++)
{
  if (ROIvector[i].empty())
  {
    break;
  }
  RGBimage=ROIvector[i];
  cvtColor(ROIvector[i],HSVimage,CV_BGR2HSV_FULL);
  cvtColor(ROIvector[i],HSLimage,CV_BGR2HLS_FULL);//obs HLS ikke HSL
  cvtColor(ROIvector[i],YCrCbimage,CV_BGR2YCrCb);

  RGBcomponentScalar=sum(RGBimage);
  HSVcomponentScalar=sum(HSVimage);
  HSLcomponentScalar=sum(HSLimage);
  YCrCbcomponentScalar=sum(YCrCbimage);

  split(RGBimage,RGBindividual);
  split(HSVimage,HSVindividual);
  split(HSLimage,HSLindividual);
  split(YCrCbimage,YCrCbindividual);

  RGBcomponentMatVector.push_back(RGBindividual);
  HSVcomponentMatVector.push_back(HSVindividual);
  HSLcomponentMatVector.push_back(HSLindividual);
  YCrCbcomponentMatVector.push_back(YCrCbindividual);

  for(int l=0;l<3;l++)
  {
    RGBcomponentValueVector[l][i]=RGBcomponentScalar.val[l]/((float)
        nColums*float(nRows));
    HSVcomponentValueVector[l][i]=HSVcomponentScalar.val[l]/((float)
        nColums*float(nRows));
    HSLcomponentValueVector[l][i]=HSLcomponentScalar.val[l]/((float)
        nColums*float(nRows));
    YCrCbcomponentValueVector[l][i]=YCrCbcomponentScalar.val[l]/((
        float)nColums*float(nRows));

  }

}

grapVectorArray(RGBcomponentValueVector[0],windowsize,"RGB comp","B")
    ;
grapVectorArray(RGBcomponentValueVector[1],windowsize,"RGB comp","G")
    ;
```

```
grapVectorArray(RGBcomponentValueVector[2],windowsize,"RGB comp","R")
    ;

grapVectorArray(HSVcomponentValueVector[0],windowsize,"HSV comp","H")
    ;
grapVectorArray(HSVcomponentValueVector[1],windowsize,"HSV comp","S")
    ;
grapVectorArray(HSVcomponentValueVector[2],windowsize,"HSV comp","V")
    ;

grapVectorArray(HSLcomponentValueVector[0],windowsize,"HSL comp","H")
    ;
grapVectorArray(HSLcomponentValueVector[2],windowsize,"HSL comp","S")
    ;
grapVectorArray(HSLcomponentValueVector[1],windowsize,"HSL comp","L")
    ;

grapVectorArray(YCrCbcomponentValueVector[0],windowsize,"YCrCb comp",
    "Y");
grapVectorArray(YCrCbcomponentValueVector[1],windowsize,"YCrCb comp",
    "Cr");
grapVectorArray(YCrCbcomponentValueVector[2],windowsize,"YCrCb comp",
    "Cb");

//obs HLS ikke HSL
if(global_sendToSwapFile==true)
{
  Mat matArray= convertVector2Mat(RGBcomponentValueVector[0]);
  outputArray(matArray,windowsize,"Blue",true);
  matArray= convertVector2Mat(RGBcomponentValueVector[1]);
  outputArray(matArray,windowsize,"Green");
  matArray= convertVector2Mat(RGBcomponentValueVector[2]);
  outputArray(matArray,windowsize,"Red");
  matArray= convertVector2Mat(HSLcomponentValueVector[0]);
  outputArray(matArray,windowsize,"HSL Hue");
  matArray= convertVector2Mat(HSLcomponentValueVector[2]);
  outputArray(matArray,windowsize,"HSL Saturation");
  matArray= convertVector2Mat(HSLcomponentValueVector[1]);
  outputArray(matArray,windowsize,"HSL Light");
  matArray= convertVector2Mat(HSVcomponentValueVector[0]);
  outputArray(matArray,windowsize,"HSV Hue");
  matArray= convertVector2Mat(HSVcomponentValueVector[1]);
  outputArray(matArray,windowsize,"HSV Saturation");
  matArray= convertVector2Mat(HSVcomponentValueVector[2]);
  outputArray(matArray,windowsize,"HSV Value");
}
else
{
  Mat matArray;
  matArray= convertVector2Mat(RGBcomponentValueVector[1]); //Green
      component
  DFTandAnalyzeMatArray(matArray,FPS,windowsize,"RBG G");
  matArray=convertVector2Mat(HSVcomponentValueVector[0]);  //Hue HSV
  DFTandAnalyzeMatArray(matArray,FPS,windowsize,"HSV H");
  matArray= convertVector2Mat(HSVcomponentValueVector[1]); //
      Saturation component
  DFTandAnalyzeMatArray(matArray,FPS,windowsize,"HSV S");
  matArray= convertVector2Mat(HSVcomponentValueVector[2]); //Value
      component
  DFTandAnalyzeMatArray(matArray,FPS,windowsize,"HSV V");
  matArray= convertVector2Mat(HSLcomponentValueVector[2]); //HLS
      Saturation component
```

```
    DFTandAnalyzeMatArray(matArray,FPS,windowsize,"HSL S");
    matArray= convertVector2Mat(HSLcomponentValueVector[1]); //HLS
        value component
    DFTandAnalyzeMatArray(matArray,FPS,windowsize,"HSL L");
  }

  try
  {
    CvPlot::RemoveFigureFFS("HSV comp");
    CvPlot::RemoveFigureFFS("RGB comp");
    CvPlot::RemoveFigureFFS("HSL comp");
    CvPlot::RemoveFigureFFS("YCrCb comp");
  }
  catch(Exception &e)
  {
    cout<<e.what()<<endl;
  }
}
```

Listing 8 – Function for splitting frames into different color spaces. Can also send data to swapfile.

B.3.2 FFT and HR selection

```
void DFTandAnalyzeMatArray(Mat matArray,int FPS,int windowsize,string
    name)
{
  Scalar meanScalar;
  meanScalar=mean(matArray);
  subtract(matArray,meanScalar,matArray);
  cv::normalize(matArray,matArray,-1,1,CV_MINMAX);
  //grapMatArray(matArray,windowsize,"Array after Normalization","Value
      ");
  hammingWindow(matArray,windowsize);
  //grapMatArray(matArray,windowsize,"Array after window","Value");
  cv::dft(matArray,matArray,DFT_SCALE);
  matArray=cv::abs(matArray);
  //grapMatArray(matArray,windowsize,"DFT",name);
  windowFunction(matArray,windowsize,FPS);
  //grapMatArray(matArray,windowsize,"DFT after window",name);

  int index=findMaxInMatArray(matArray,windowsize);
  float HR=calculateHR(windowsize,FPS,index);

  cout<<"Name= "<<name<<" HR= "<<HR<<endl;
  outputHR(HR,false);
  idft(matArray,matArray,DFT_SCALE);
  //grapMatArray(matArray,windowsize,"IDFT",name);
}
```

Listing 9 – Function that inputs an single array, does signal processing and outputs HR frequency

B.3.3 Hamming window

```
void hammingWindow(Mat src,int length)
```

```
{
  Mat hammingWindowMat(length,1,CV_32FC1);
  for (int i=0;i<length;i++)
  {
    hammingWindowMat.at<float>(i)=0.5*(1-cos((2*PI*i)/(length-1)));
  }
  cv::multiply(src,hammingWindowMat,src);
}
```

B.3.4 Window function

```
void windowFunction(Mat array, int windowsize,int FPS)
{
  float lowCutFreq=0.75;//Hz eq. HR 45
  float highCutFreq=3.33;//Hz eq. HR 200
  float freqStep=float((FPS/2))/float(windowsize);

  int lowCutofIndex=floor(lowCutFreq/freqStep);
  int highCutofIndex=ceil(highCutFreq/freqStep);
  for(int i=0;i<windowsize;i++)
  {
    if(i<lowCutofIndex ||i>highCutofIndex)
    {
      array.at<float>(i)=0;
    }
  }
}
```

Listing 10 – Simple vindow function that removes frequencies outside HR frequency range

C Other

C.1 Technical data

C.1.1 Pulse Oximeter Specifications

Contec CMS50E, Contec Medical Systems Co., Ltd.

Main Parameters (from user manual)

Measurement of SpO_2 Measurement range: 0-100%, accuracy: 70-100%, ±2%; 0-69%, unspecied.

Measurement of pulse rate Measurement range: 30-250 bpm, accuracy: ±2 bpm or ±2% (select larger).

Resolution SpO_2: 1%, pulse rate: 1 bpm.

Resistance to surrounding light The deviation between the value measured in the condition of man-made light or indoor natural light and that of dark room is less than ±1%.

Optical sensor Red light (wavelength is 660 nm, 6.65 mW). Infrared (wavelength is 880 nm, 6.75 mW).

C.1.2 Web Camera Specifications

Microsoft LifeCam Studio

Imaging features(from technical data sheet)

Sensor CMOS sensor technology

Resolution Sensor resolution: 1920×1080, video resolution: up to 1920×1080.

Field of view 75^0 diagonal field of view.

Frame rates Supported frame rates are 7.5, 10, 15, 20 and at most 30 frames per second, depending on resolution.

Other Digital only pan, tilt and zoom. Auto focus from 10 cm. Automatic image adjustment with manual override.

C.1.3 Sony PMW-EX1 Camera Specification

Sony PMW-EX1 (Data from data sheets)

Focal Length f=5.8 to 81.2 mm

Zoom Ratio 14x

Resolution Up to 1920x1080

Sensor 3-chip 1/2-inch Exmor CMOS

Frame rate Up to 60 FPS

Media format XDCAM