DISSERTAÇÃO DE MESTRADO

AUTOMATED NON-CONTACT HEART RATE MEASUREMENT USING CONVENTIONAL VIDEO CAMERAS

Gustavo Luiz Sandri

Brasília, Fevereiro de 2016

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

AUTOMATED NON-CONTACT HEART RATE MEASUREMENT USING CONVENTIONAL VIDEO CAMERAS

Gustavo Luiz Sandri

Dissertação de Mestrado submetida ao Departamento de Engenharia

Elétrica como requisito parcial para obtenção

do grau de Mestre em Engenharia de Sistemas Eletrônicos e Automação

Banca Examinadora

Prof. Ricardo Lopes de Queiroz, Ph.D., CIC/UnB
Orientador
_____

Prof. João Luiz Carvalho, Ph.D., ENE/UnB
Examinador Interno
_____

Prof. Alexandre Zaghetto, Dr., CIC/UnB
Examinador Externo
_____

Prof. Daniel Guerreiro e Silva, Dr., ENE/UnB
Examinador Suplente
_____

FICHA CATALOGRÁFICA

REFERÊNCIA BIBLIOGRÁFICA

SANDRI, G.L. (2016). *AUTOMATED NON-CONTACT HEART RATE MEASUREMENT USING CONVENTIONAL VIDEO CAMERAS*. Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 70 p.

CESSÃO DE DIREITOS

_____

Gustavo Luiz Sandri

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Acknowledgments

ABSTRACT

As the blood flows through the body of an individual, it changes the way that light is irradiated by the skin because blood absorbs light differently than the remaining tissues. This subtle variation can be captured by a camera and be used to monitor the heart activity of a person.

The signal captured by the camera is a wave that represents the changes in skin tone along time. The frequency of this wave is the same as the frequency by which the heart beats. Therefore, the signal captured by the camera could be used to estimate a person's heart rate. This remote measurement of cardiac pulse provides more comfort as it avoids the use of electrodes or others devices attached to the body.

In this work we propose two algorithms for non-contact heart rate estimation using conventional cameras under uncontrolled illumination. The first proposed algorithm is a simple approach that uses a face detector to identify the face of the person being monitored and extract the signal generated by the changes in the skin tone due to the blood flow. This algorithm employs an adaptive filter to boost the energy of the interest signal against noise. We show that this algorithm works very well for videos with little movement.

The second algorithm we propose is an improvement of the first one to make it more robust to movements. We modify the approach used to define the region of interest. In this algorithm we employ a skin detector to eliminate pixels from the background, divide the frames in micro-regions that are tracked using an optical flow algorithm to compensate for movements and we apply a clustering algorithm to automatically select the best micro-regions to use for heart rate estimation. We also propose a temporal and spatial filtering scheme to reduce noise introduced by the optical flow algorithm.

We compared the results of our algorithms to an off-the-shelf fingertip pulse oximeter and showed that they can work well under challenging situations.

RESUMO

Conforme o sangue flui através do corpo de um indivíduo, ele muda a forma como a luz é irradiada pela pele pois o sangue absorve luz de forma diferente dos outros tecidos. Esta sutil variação pode ser capturada por uma câmera e ser usada para monitorar a atividade cardíaca de uma pessoa.

O sinal capturado pela câmera é uma onda que representa as variações de tonalidade da pele ao longo do tempo. A frequência desta onda é a mesma frequência na qual o coração bate. Portanto, o sinal capturado pela câmera pode ser usado para estimar a taxa cardíaca de uma pessoa. Medir o pulso cardíaco remotamente traz mais conforto pois evita o uso de eletrodos.

Neste trabalho nós propomos dois algoritmos para a estimação da taxa cardíaca sem contato usando câmeras convencionais sob iluminação não controlada. O primeiro algoritmo proposto é um método simples que emprega um detector de face que identifica a face da pessoa sendo monitorada e extrai o sinal gerado pelas mudanças no tom da pele devido ao fluxo sanguíneo. Este algoritmo emprega um filtro adaptativo para aumentar a energia do sinal de interesse em relação ao ruído. Nós mostramos que este algoritmo funciona muito bem pra vídeos com pouco movimento.

O segundo algoritmo que propomos é uma melhora do primeiro para torná-lo mais robusto a movimentos. Nós modificamos o método usado para definir a região de interesse. Neste algoritmo é utilizado um detector de pele para eliminar píxeis do plano de fundo do vídeo, os frames dos vídeos são divididos em micro-regiões que são rastreados com um algoritmo de fluxo ótico para compensar os movimentos e um algoritmo de clusterização é aplicado para selecionar automaticamente as melhores micro-regiões para efetuar a estimação da taxa cardíaca. Propomos também um esquema de filtragem temporal e espacial para reduzir o ruído introduzido pelo algoritmo de fluxo ótico.

Comparamos os resultados dos nossos algoritmos com um oxímetro de dedo comercial e mostramos que eles funcionam bem para situações desafiadoras.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Symbols

AC	As in Alternating Current, represents the signal component of frequency different than zero
DC	As in Direct Current, represents the signal component of frequency zero
f_s	Sampling frequency of the input video, in frames per second
T	Window duration
ΔT	Window time increment
$I[i]$	i-th frame of the input video
N_{FT}	Number of elements used to compute the DFT
N_T	Number of frames within T seconds
$N_{\Delta T}$	Number of frames within ΔT seconds
$p[j]$	j-th estimated Heart Rate
$x_r[i]$; $x_g[i]$; $x_b[i]$	The i-th red, green and blue traces, respectively, computed by the average value of the pixels inside the ROI, computed from the corresponding color channel
$y_r[j,k]$; $y_g[j,k]$; $y_b[j,k]$	The k-th normalized red, green and blue traces, respectively, on the j-th time window
$Z[j,v]$	Discrete Fourier Transform of the normalized trace on the j-th time window
$Z_f[j,v]$	Filtered Discrete Fourier Transform of the normalized trace on the j-th time window by the adaptive filter
$|\epsilon|$	Absolute error between estimate heart rate and oximeter reading
σ_1, σ_2	Parameters of the bilateral filter
σ_s	Parameter of the similarity measure

Acronyms

BCG	Ballistocardiograph
BPM	Beats per Minute
BSS	Blind Source Separation
CIE	*Commission Internationale de l'Eclairage*
DFT	Discrete Fourier Transform
DRMF	Discriminative Response Mat Fitting
ECG	Electrocardiogram
FFT	Fast Fourier Transform
FPS	Frames per Second
HR	Heart Rate
ICA	Independent Component Analysis
MLP	Multi Layer Perceptron
PCA	Principal Component Analysis
PPG	Photoplethysmography or Photoplethysmographic
RMS	Root Mean Squared
ROI	Region of Interest
SNR	Signal to Noise Ratio
SOM	Self Organizing Map
SpO_2	Arterial Oxygen Saturation
STFT	Short Time Fourier Transform
HR-FD	Video heart rate estimation through Face Detection
HR-MRT	Video heart rate estimation through micro-region tracking
USB	Universal Serial Bus

Color Spaces

R	Red
G	Green
B	Blue
H	Hue
T	Tint
S	Saturation
V	Value
L	Lightness
I	Intensity
Y	Luma
Cr	Chrominance red
Cg	Chrominance green
Cb	Chrominance blue

1 INTRODUCTION

The heart contracts rhythmically to drive nutrients and oxygen necessary for life to our cells. The heart rate, that is the frequency by which the heart is beating, is a measure that can be used to verify a person's health and emotions.

As the heart beats, several characteristics of the body changes along the beating and can be employed to detect someones heart rate, such as electrical activity, blood pressure and light absorption by the skin. Electrocardiography (ECG) is the most precise and traditional method to detect heart rate and some anomalies [1, 2]. It records the electrical activity due to the heart muscle depolarization in each heartbeat using electrodes placed on the patient skin. One disadvantage of this technique is its need to use electrodes and equipment specially build for this purpose, making it harder for the everyday usage.

An alternative for the ECG is to use photoplethysmography (PPG) [3], a technique that uses light-based technology to captured the changes in skin light absorption as the blood flows. This is possible because the blood absorbs light differently than the others tissues of the skin. Hence, as the density of blood beneath the skin changes, as a result from the heartbeats, the total light absorption of the skin changes accordingly. Thus, methods for heart rate measurement that employ light as a signal measures the frequency by which the light emitted by the skin changes and attribute this frequency as the heart rate.

To capture the PPG signal one can use a photodiode to produce a controlled light, place it near the skin and capture, with a photoreceptor device, either the light that traversed the skin or the one that was reflected by it. These devices are commonly called pulse oximeters [4]. In a less controlled environment, it is possible to avoid the use of a photodiode and employ the ambient light as a light source. The disadvantage is that, as the ambient light is hard or, sometimes, impossible to control, we do not know precisely which wavelengths are actually used and their magnitude. The advantage is that, without the used of a controlled light source, we can perform a non-contact heart rate estimation using conventional cameras that are widely available. In this fashion, we trade precision in heart rate estimation for comfort, as we avoid the use of electrodes and/or sources of lights placed on the skin, and the possibility of non-contact estimation that enable us to monitor a patient for long time periods or even to monitor someone in a concealed way.

For heart rate estimation using videos, most methods focus their attention on the face of the person being monitored because it has been shown that the face is a part of the body where the changes in skin absorption is sufficiently high to be perceived by a camera, even under natural light [5]. This is due to the fact that the skin in the face is more vascularized near the surface than the rest of the body. From a video, a region of interest comprising the face of the person being monitored is defined and the mean value of red, green and blue inside the region of interest is

1

computed. These mean values are commonly called as red, green and blue traces. The heart rate is then estimated examining how this traces changes with time.

1.1 GOALS

This work aims to develop an algorithm to estimate the heart rate of human beings based on videos captured from their face under ambient light of indoors enviroments. The algorithm should be robust to noise and movements and capable of performing the estimation without supervision.

1.2 CONTRIBUTIONS

A number of algorithms have been proposed to estimate the heart rate using a video camera [6, 7, 8, 9, 10, 5, 11, 12, 13, 14, 15]. Our work builds on top of these algorithms, modifying some key aspects in order to improve the overall reliability of the system. The main contributions of this work are as follows:

- Proposed an adaptive filter in order to make the algorithm perform better when facing noise;

- Modified how the red, green and blue traces are combined, aiming to increase the number of correct estimated heart rate. Most algorithms in the literature employ blind source separation [10, 16, 11, 13, 14] or only the green channel [7, 8, 5, 12]. In this work we proposed a fixed mixture of the three signals avoiding to explicit blind separate them. We justify why this approach is similar to blind separation of the signals while avoiding artifacts that could be introduced by the blind source separator when it is not able to correct separate the sources;

- Proposed a method to compensate for movements while capturing the red, green and blue traces from the face. We divided the first frame of a time block of the video in micro-regions and track some points of them to determined how the micro-regions evolved with time. The information of the tracking points are temporally filtered using a polynomial model and spatial filtered using an affine transformation;

- Implemented a clustering algorithm to automatically selected the best micro-regions to be used for heart rate estimation. Thus, the algorithm chooses the region of interest automatically having an arbitrary shape.

We also showed where are the best regions on the human face for heart rate estimation. Also, the proposed algorithm for point tracking filtering is new and could be used for other purposes.

1.3 PRESENTATION OF THE MANUSCRIPT

Chapter 2 presents a review of the literature. We explain the Photoplethysmographic signal, its bases and why we are able to capture it and estimate a person's heart rate using a conventional camera. We review skin detection algorithms that we employ in our work to define the region of interest and we present an overview of other algorithms in the literature that estimated the heart rate using conventional cameras.

Chapters 3 and 4 disclose the proposed algorithms. In Chapter 3 we describe the first proposed algorithm, introducing an adaptive and a derivative filter and how the signal from the three color channels are combined to enhance the SNR. We also present evidences that this algorithm do not perform very well in presence of movement. Therefore, Chapter 4 describes the second proposed algorithm, an improvement of the first to make it robust to movements. In this algorithm we divide the frames in micro-regions and use point tracking and filtering to compensate for movements. We also present the clustering algorithm used to select micro-regions for heart rate estimation, automatically defining the region of interest.

Finally, Chapter 5 presents the results obtained with our algorithms, compared to the literature, for real and synthetic signals and Chapter 6 presents the conclusion of this work.

2 LITERATURE REVIEW

This chapter address the main topics found in the literature related to our work. A complete review is out of the scope of this manuscript, hence, we concentrate our attention to explain the ideas upon which this work is based. Further literature is also found in subsequent chapters.

Section 2.1 is an introduction to the biological concepts that explain how one can estimate the heart rate using video cameras. Section 2.2 is dedicated to explain skin detection algorithms that are commonly used for heart rate detection based on the fact that the interest regions of the videos is the subject skin. In section 2.3 we present the work of other authors in the domain of heart rate detection using conventional cameras.

2.1 PHOTOPLETHYSMOGRAPHY

The signal of interest in our work is the wave produced by blood flowing through the body, known as the photoplethysmographic (PPG) waveform [17]. With each heart beat, a pressure pulse radiates out to the peripheral circulatory system causing significant change in the arterial and capillary diameters [18, 3], thus increasing the volume of blood beneath the skin. As pressure decreases, arteries return to their normal size.

Blood possesses a great amount of red cells, responsible for oxygen transportation. These cells contain hemoglobin, that absorbs light differently from others structures of epithelial tissue [19]. Skin tissue has a relatively low absorptivity for wavelengths in the visible and near-infrared light spectrum (400–2000 nm). This characteristic is mainly due to skin pigments (particularly melanin) and water [20]. On the other hand, the absorption for the hemoglobin varies depending whether it is loaded or not with oxygen or carbon monoxide [19]. Two of these variants are of interest in this work: the oxyhemoglobin and de-oxyhemoglobin.

The oxyhemoglobin ($Hb0_2$) is formed when oxygen binds to hemoglobin in red blood cells during physiological respiration, while de-oxyhemoglobin (Hb), or deoxygenated hemoglobin, is the form of hemoglobin without the bound oxygen [19]. Figure 2.1 shows their absorption spectra. Oxyhemoglobin has significantly lower absorption for green light at 560 nm and a higher absorption for blue light at 480 nm.

As blood flows, the density of oxyhemoglobin and de-oxyhemoglobin underneath the skin changes periodically. The amount of red cells near the surface of skin also changes, which alters the average distance that light must travel before being reflected. Light, while traveling through the skin tissue, interacts with it resulting primarily in reflection and absorption, although scattering, transmission and fluorescence may also be present [21]. This affects the way that skin radiates back the environment light.

4
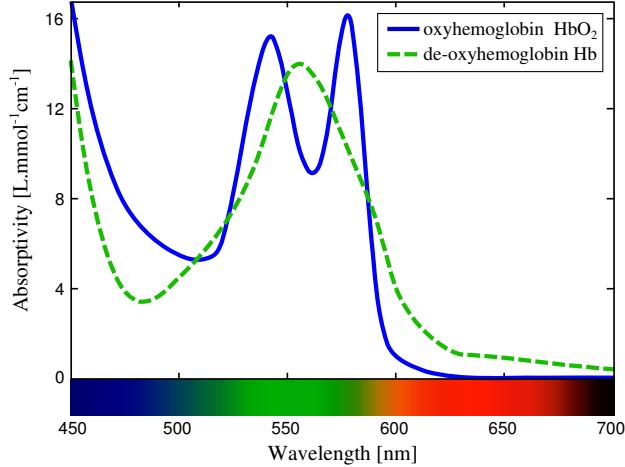
Figure 2.1: Millimilar absorptivity spectra of hemoglobin in the visible range. Adapted from Zijlstra *et al.* [19]

The most traditional way to measure the photoplethysmographic signal is to use an infrared light source (usually a photodiode) to illuminate the skin and a photodetector placed either in the opposite side, to capture the transmitted light, or on the same side, to capture the reflected light [17] (see Figure 2.2). The main peripheral sites where the PPG can be obtained are fingers' tissue pads, ears and toes where there is a high degree of superficial vasculature [22].
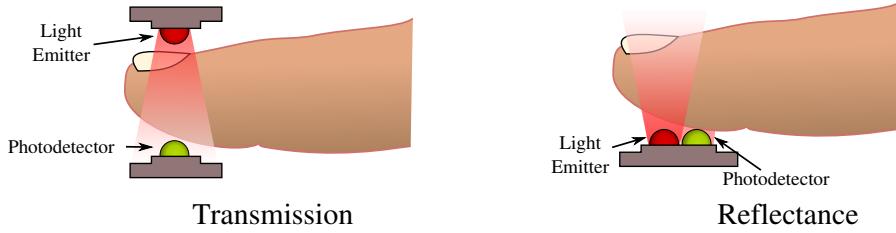
Figure 2.2: Measurement of PPG signal using infrared light in contact with finger. Either the transmitted or reflected light is used, captured by a photodetector.

The changes on the skin absorption due to blood circulation is a phenomenon that has been known for a long time [4]. It was first described by Alrick Hertzman in 1937 [23]. He believed, based on his observations, that the origin of the PPG signal was linked to blood volume changes. Therefore, he named it as "photoelectric plethysmograph". The term "plethysmos" derives from the Greek word for fullness and expressed his belief that he was measuring the fullness of the tissue when he measured the amount of light absorption. Later researches demonstrated that he was not far on his assumption, with results for the PPG being close to the more traditional strain gauge plethysmograph [24], a method that provides a quantitative measure of the bloodflow.

Electrocardiogram (ECG) and PPG signals share some characteristics, as both of them originate from the heart beats. For example, they have the same fundamental frequency (the heart rate) and the systole and diastole are visible on both of them [17]. However, their waveform differs significantly, as it can be seen in Figure 2.3, which is related to the way how each wave travels through the body and how it is measured. Also, one can notice a phase difference between them due to the fact that electric waves travel much faster than pressure waves, originating different

delays.
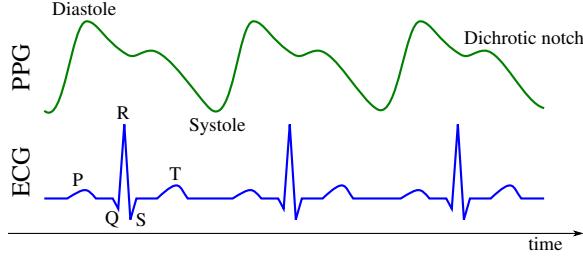
Figure 2.3: Schematic comparing PPG and ECG signals.

More recently, it has been shown that the PPG signal can be estimated from videos captured with a commercial camera filming the face of the subject being monitored [13, 25, 26, 27, 8, 11]. Conventional cameras divide the input light into three channels using a color filter [28]. As the PPG signal results in a periodic variation of the reflected light spectrum, this signal can be appreciated in all three color channels multiplied by a constant that is intrinsic to the light wavelengths gathered on each channel of the camera. Therefore, each channel will receive an attenuated version of the PPG signal, added to noise coming from movement artifacts, sensor temperature (depending on the sensor quality), among other noise sources.

Wu *et al.* [27] used conventional cameras and Eulerian Video Magnification to make the small changes in the skin color, due to blood circulation, visible to the naked eye. However, they did not try to detect the subject pulse, only amplify it, but their works shows the feasibility of contactless estimate the heart rate using conventional cameras. Later, they presented the Phase-Based Video Motion Processing [29], a modification of the previous algorithm, dedicated to amplify exclusively the motion. This new algorithm can be employed in the previous algorithm to make it more robust to subtle movements, resulting in a better outcome. But this approach only works when the movements are of low to very low amplitudes.

Among the image processing tools, skin detection stands out as it is used by many techniques. A detailed description of skin detection techniques is presented next, and a more thorough review of works in heart rate detection using cameras is given in Section 2.3.

2.2 SKIN DETECTION

Skin detection is an important tool in heart rate estimation through videos as it can segment the skin regions, where the PPG signal is likely to be found, from the rest of the scene. But it is also applied to a large range of other applications, including face detection [30], content-based image retrieval [31], and nudity detection [32].

The best performance for skin detection is acquired using the visual and non-visual spectrum of light, for-instance infrared [33, 34]. However, the majority of researches concentrate their effort only in visible light, given the fact that non-visible spectrum information is usually not

6

available (or too expensive to capture) and the wide use of conventional cameras that only capture red, green and blue wavelengths.

Therefore, they must cope with problems that arise with this limitation, mainly illumination, ethnic group and camera characteristics, that influence the skin color [35]. Illumination is a problem because skin detectors don't assume a controlled environment and should deal with indoor and outdoor scenes, shadows, highlights, etc. Skin color varies from white, yellow, reddish to dark accordingly to the subject ethnicity [36]. Finally, the camera, even under the same illumination, may present different resulting images depending on sensor sensitivity and color filter. There are also some other minor factors that may influence the detection, such as makeup, motion blur, etc.

The simplest skin detector are the pixel-based. They classify each pixel as skin or non-skin independently from its neighborhood [36]. In this case, the crucial issue is to choose the color space more suitable to solve the problem. Surprisingly, many papers related to skin detection do not provide a strict justification of their choice [36]. Just some few works were devoted to do a comparative analysis of different color spaces used for skin detection [37, 38, 39, 40].

We dedicate Section 2.2.1 to review the most employed color spaces for skin detection. A summary of the main techniques used for skin detection is presented from Section 2.2.2 to 2.2.5.

2.2.1 Color Spaces

The performance of a given color space for skin detection is related to how well they can separate the skin pixels from non-skin pixels and is measured by the classification error obtained on a test data [35]. It has been observed that skin colors vary more in intensity (or luminance) than in chrominance [41]. Therefore, it has become a common practice to drop the luminance component in order to add robustness to the classification when facing different illumination and ethnicity.

We present below the main color spaces exploited for skin detection.

2.2.1.1 RGB

RGB originated from cathode ray tube display applications and it is most commonly used for storing and representing digital images since the human eye naturally captures the three primary colors used for image representation (red, green and blue) which gave the name to this color space [42]. The RGB representation was standardized in 1930 by the *Commission Internationale de l'Eclairage* (CIE), using the primary colors of red (700.0 nm), green (546.1 nm) and blue (435.8 nm) [28].

However, these three channels are highly correlated and its dependencies with illumination make this space not a favorable choice [36]. Even though, some researchers have successfully employed this color space for the purpose of skin detection, avoiding the conversion of the colors

to another space, such as Brand and Mason [43] and Jones and Rehg [44].

To overcome this problem we can perform a simple normalization on the R, G and B components, resulting in the normalized RGB (or simply rgb):

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B} \quad \text{and} \quad b = \frac{B}{R+G+B}, \tag{2.1}$$

where R, G and B are the levels of red, green and blue, respectively. We can reduce the space dimensionality dropping the third component, as it becomes redundant after the normalization.

It has been reported that the normalized RGB space is more robust to skin color changes due to lighting and ethinicity and the clusters in rgb space present a lower variance than the corresponding cluster in the RGB space [41, 45].

The ratio between the colors in the RGB space is also used to detect the presence of skin. It was observed that skin contains a significant high level of red, independent of ethnicity [46]. Therefore, the R/G ratio has been used for skin detection [46]. Others ratios (R/B and G/B) were tested for skin detection by Brand and Mason [43].

2.2.1.2 Perceptual Color Spaces (HSI, HSV and HSL)

Developed in the 1970s for computer graphics applications, hue-saturation color spaces were introduced to numerically represent the artistic idea of tint, saturation and tone [47]. The dominant color (red, yellow, purple, etc.) is represented by the Hue (or Tint) while saturation represents how pure the color is, starting with gray when the saturation is zero and going up to pure color when the saturation is maximum. The third component (Intensity, Value or Lightness) measures how bright the color is.

They cannot be directly described by the RGB space, but many non-linear transformations were proposed to relate them with the RGB space, for example [36]:

$$H = cos^{-1} \left(\frac{1}{2} \frac{(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right) \tag{2.2}$$

$$S = 1 - 3\frac{min(R,G,B)}{R+G+B} \tag{2.3}$$

$$V = \frac{R+G+B}{3} \tag{2.4}$$

These transformation are invariant to highlights at white lights, ambient light and orientation relative to light sources, making it a good choice for skin detection [28]. Another color space similar to the hue-saturation is the normalized chrominance-luminance TSL space, that is a transformation of the normalized RGB into more intuitive values, close to hue and saturation in their meaning.

2.2.1.3 YCbCr

YCbCr is an orthogonal color space, commonly used for image compression, that uses statistical independent components aiming to reduce the redundancy of RGB [28]. Color is represented by *luma* (Y), which is the luminance computed from a weighted sum of RGB values, and two chrominances: chrominance blue (Cb) and chrominance red (Cr) that are computed by subtracting *luma* from red and blue components.

$$Y = 0.299R + 0.587G + 0.114B \tag{2.5}$$

$$C_b = B - Y \tag{2.6}$$

$$C_r = R - Y \tag{2.7}$$

The explicit separation between luminance and chrominance and its simplicity makes YCbCr one of the most popular choices for skin detection. Other similar color spaces, such as YCgCr, YIQ, YUV and YES, derived from YCbCr, are also employed.

2.2.1.4 Psychophysical Based Spaces

The RGB space is not a perceptually uniform color space, meaning that the same amount of variation in the values of red, green and blue, applied at two different levels, will not be perceived the same way by the human brain. Therefore, the CIE, based on psychophysical experiments, introduced the CIELAB and CIELUV that try to match the characteristics of human visual system [48]. The price for better perceptual uniformity is complex transformation functions from and to RGB space.

2.2.2 Color Thresholding

Given a color space, the components of skin pixels from different individuals tend to cluster in a small region [35]. Hence, one simple method to discriminate skin pixels is to explicitly define the boundaries of the skin cluster. One or more color spaces can be chosen and a pixel is classified as skin only if its parameters fall within all predetermined ranges.

The obvious simplicity of this method and its easy implementation, that leads to fast computation, has attracted many researchers [49, 50, 51, 52]. The main difficulty of this approach is the need to empirically select the decision boundaries that is strongly dependent on the chosen color space. So, both a good color space and good decision boundaries must be found.

Table 2.1 presents examples of thresholds employed for skin detection as proposed by some authors.

Table 2.1: Examples of skin discrimination using explicit thresholding.

Authors		Color Space	Boundaries
Peer *et. al.*	[49]	RGB	$R > 90,\ \ G > 40,\ \ B > 20,\ \ R > G + 15,\ \ R > B,$ $max(R, G, B) - min(R, G, B) > 15$
Tsekeridou and Pitas	[50]	HSV	$V \geq 40,\ \ \ 0.2 < S < 0.6,$ $0° < H < 25°\ \ \text{or}\ \ 335° < H < 360°$
Chai and Ngan	[51]	YCbCr	$77 \leq Cb \leq 127,\ 133 \leq Cr \leq 173$
Khan *et. al.*	[52]	CIELAB	$2 \leq A \leq 14,\ \ \ 0.7 \leq B \leq 18$

2.2.3 Histogram Based Approach

This method is used to produce a probabilistic classifier. It employs a database of previously segmented image pixels into two groups: skin and non-skin; and computes a 3D or 2D color histogram. In the case of 2D histograms, the brightness component is dropped out. The pixel components are quantized into a number of histogram bins that stores the number of times the given bin color occurred in the test data [36].

Two histograms are computed, one for skin pixels and another for non-skin pixels. These histograms are then normalized by their total sum for producing an estimation of the $P(c|skin)$ and $P(c|\overline{skin})$ (where c is a color column vector) that represent the probability of occurrence of c, given that it is a skin pixel and a non-skin pixel, respectively.

Finally, with the estimated probabilities, a lookup table is computed assigning that a bin is a skin pixel whenever its probability of occurrence is higher in the skin class than in the non-skin class.

For a more complete representation, one can use a Bayes classifier, that classifies a pixel as skin when

$$\frac{P(skin|c)}{P(\overline{skin}|c)} > \Theta, \tag{2.8}$$

for a given detection threshold Θ. $P(skin|c)$ and $P(\overline{skin}|c)$ can be computed from $P(c|skin)$ and $P(c|\overline{skin})$, resulting in

$$\frac{P(c|skin)P(skin)}{P(c|\overline{skin})P(\overline{skin})} > \Theta, \tag{2.9}$$

where $P(skin)$ and $P(\overline{skin})$ are estimated from the database.

2.2.4 Gaussian Model

The cluster formed by skin pixels in the chosen color space can be modeled by a multivariate Gaussian distribution [41], defined as:

$$P(c|skin) = \frac{1}{2\pi\sqrt{|C|}}exp\left(-\frac{1}{2}(c-\mu)^T C^{-1}(c-\mu)\right),$$ (2.10)

where μ is the mean value and C the covariance matrix of the cluster. Let c_i be the parameters of i-th skin pixel in the database and N the total number of skin pixels, then:

$$\mu = \frac{1}{N}\sum_i c_i$$ (2.11)

$$C = \frac{1}{N-1}\sum_i (c_i-\mu)(c_i-\mu)^T$$ (2.12)

After estimating the parameters μ and C, $P(skin|c)$ is used to calculate the likelihood of pixel c to be a skin pixel and the classification is performed by means of thresholding this value.

To represent a more complex-shaped cluster, a weighted sum (also called mixture) of Gaussians can be employed in the form [53]:

$$P(c|skin) = \sum_j \frac{w_j}{2\pi\sqrt{|C_j|}}exp\left(-\frac{1}{2}(c-\mu_j)^T C_j^{-1}(c-\mu_j)\right),$$ (2.13)

where w_j is the weight attributed to the j-th Gaussian and μ_j and C_j are its parameters.

2.2.5 Artificial Neural Networks

Artificial intelligence approaches have been successfully applied to skin detection due to their ability to represent complex non-linear relationship between inputs and outputs [54]. The two most used are the Self Organizing Map (SOM), devised by Kohonen, that is based on an unsupervised and competitive algorithm, and the multi layer perceptron (MLP), a simple feed forward network [36].

In the MLP, the weights in each neuron are updated iteratively using a training set for which we know the correct outcome (usually human defined). At each iteration during the training process, all inputs are processed in a feed-forward fashion and the output is compared with the expected result. Through a gradient descend technique, the error is back-propagated to adjust the neurons weights.

The SOM, on the other hand, constructs a topological structure that represents high dimensional data in a fashion that allows relative distances to be preserved. During the training phase, the weights of each neuron are updated to become similar to the input data, and neighboring neurons will have similar weights, thus creating a clustering.

Neural networks have been shown a good performance as they can easily generalize complex structures [55]. However, one disadvantage is its dependence on several factors, such as the number of hidden nodes and the training. The network must be extensively tuned in order to obtain an optimal performance.

2.3 PULSE DETECTION

Detection of heart rate using cameras has the advantage of being a non invasive technique. For example, it enables to monitor, in comfort, a patient for a long time or to monitor a subject in a concealed manner, where knowing that he is being monitored wouldn't be desired, like in a lie detector. However, when dealing with heart rate detection with cameras one must take into account others sources of errors, like movements, uncontrolled illumination, etc.

There are two main approaches to estimate the heart rate of a subject using conventional cameras without contact. The first is the pixel based approach where the estimation is based on how skin pixel change their tone with time [7, 8, 13]. The second one is based on micromovements of the body due to the heart beats that are detected using a feature tracking [56].

2.3.1 Pixel Based

In the pixel based approaches, a conventional camera films the subject being monitored and the PPG signal is captured by three color channels, red, green and blue. One or multiple regions of interest (ROI) are defined and the average value for the red, green and blue, within the ROI, is computed over time. These three signals are then employed to estimate the heart rate, either in time or frequency domain. Table 2.2 is a summary of pixel based algorithms employed in the literature.
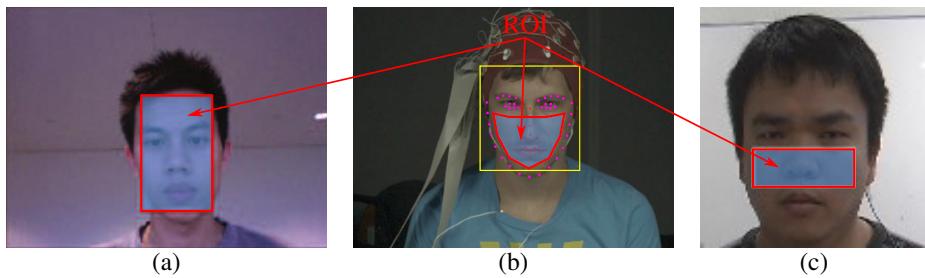
Figure 2.4: ROI, represented in red, as defined by (a) Poh *et al.* [13], (b) Li *et al.* [8] and (c) Yu *et al.* [10]. Adapted.

Most algorithms, such as those of Poh *et al.* [13, 14], Kwon *et al.* [16] and Purshe *et al.* [11], uses a cascade classifier to detect the subject face as they have shown that the PPG signal is relatively easy to detect on the face skin. Capdevila *et al.* [5] have shown that the forehead, cheeks and chin are the best regions on the face to measure the PPG signal. The ROI is defined as a rectangle comprising most of the subjects face (see Figure 2.4).

Table 2.2: Summary of algorithms employed in the literature for HR detection with video.

Authors		ROI	Camera	Signal employed[1]
Ufuk Bal	[6]	face[2] (Viola Jones + skin detection)	Webcam; 640x480; 30 FPS	fixed mixture of R, G and B, denoised and band-pass filtered
Couderc *et al.*	[7]	face[3] (entire frame)	Microsoft LifeCam Cinema; 2MP; 15 or 30 FPS	green channel band-pass filtered and interpolated
Li *et al.*	[8]	face[4] (DRMF)	IPAD (built-in iSight camera); 640x480; 30 FPS	green channel band-pass filtered + Welch periodogram
Xu *et al.*	[9]	manual selection[3]	Conventional camera and smartphones (using a dermatoscope to amplify the skin 20x)	derivative of $log(R/G)$ band-pass filtered + DFT
Balakrishnan *et al.*	[56]	face[4] (Viola Jones)	Panasonic Lumix GF2; 1280x720; 30 FPS	micro-movements (vertical) band-pass filtered + PCA
Yu *et al.*	[10]	face[2]	conventional camera; 720x576; 25 FPS	ICA + STFT
Capdevila *et al.*	[5]	face[3] (manual)	Canon Ixux 80is; 640x480;	green channel band-pass filtered and interpolated
Kwon *et al.*	[16]	face[2] (Viola Jones)	smartphone (iPhone 4); 640x480; 30 FPS	ICA + DFT
Pursche *et al.*	[11]	face	webcam; 1.3 MP; 30 FPS	ICA + DFT
Bolkhovsky *et al.*	[12]	right index finger[3] (covering the lens)	smartphones; 30 or 20 FPS	green channel interpolated + Welch periodogram
Poh *et al.*	[13, 14]	face[2] (Viola Jones)	Macbook Pro (built-in iSight camera); 640x480; 15 FPS	ICA + DFT
Verkruysse *et al.*	[15]	manual selection[3]	Canon Powershot A560; 640x480 or 320x240; 15 or 30 FPS	R, G, and B channels (individually) + DFT

[1] Describes how the red, green and blue traces computed on the ROI are used to estimate de HR. Most of them remove the DC component from the traces. Thus this step was omitted.

[2] Reset at each frame

[3] Fixed region over time

[4] Updated through point tracking

Li *et al.* [8] employs Discriminative Response Mat Fitting (DRMF) [57], a more robust face fitting algorithm, that estimates a set of parameters describing the position of the eyes, mouth, chin, nose and eyebrows. They define the ROI as the region comprising mainly the cheeks, as shown in Figure 2.4.

From the averages values computed on the ROI, one for each color channel, the researches either use only the green channel based on the fact that the PPG wave is more strongly present in this channel [15] or a combination of the three by means of Independent Component Analysis (ICA), Principal Component Analysis (PCA) or with fixed weight, linearly mixing them in an attempt to maximize the Signal to Noise Ratio (SNR) for the PPG signal. ICA is the preferred approach among researchers. Band-passing the signal to eliminate those frequencies that do not correspond to the heart rate (HR) is also a common practice.

Xu *et al.* [9], on the other hand, compute the discrete derivative of the logarithmic ratio between red and green traces. The use of the logarithmic function is explained making use of the Lambert-Berr law, where the signal received by each color channel can, approximately, be said to be proportional to $e^{-(\nu(\lambda)\rho(t)+A_0(\lambda))}$, where $\nu(\lambda)$ and $A_0(\lambda)$ are the absortivity of hemoglobin multiplied by the mean path that light travels before being reflected and the absorbance of others tissues in the skin, respectively, for a given wavelength λ. The concentration of hemoglobin is given by $\rho(t)$, and variate with time t. The discrete time derivative is used to eliminate DC components and to attenuate low frequency noises.

The signal is converted to the frequency domain using Discrete Fourier Transform (DFT) [16, 11, 14, 13, 15], Short Time Fourier Transform (STFT) [10] or Welch periodogram [8, 12] (a method used to estimate the power spectra [58]), and the frequency corresponding to the peak of maximum power is attributed as the heart rate (HR) frequency. The search of the HR is limited to a range of values where they expect the pulse frequency to be.

Some researches also use an approach in the time domain. Couderc *et al.* [7], for example, first applies a band-pass filter to the signal to eliminate those frequencies that do not correspond to the pulse and cubic interpolate the signal. Then, they search for the position of the peaks and valleys and the HR frequency is computed as the inverse of the signal period. The problem with this approach is that it degrades rapidly with noise.

2.3.2 Algorithm of Poh

The work we developed here is based on the algorithm of Poh *et al.*. Many of the related works in this field follow a similar structure (or are based in Poh algorithm) as can be seen in Table 2.2. Therefore, we explain it in more detail here. Figure 2.5 presents a schematic of their algorithm.

Let $I[i]$ be the i-th frame of the input video. For each frame, a cascade of boost classifier that uses 14 Haar-like features trained with positive and negative examples of frontal faces, based on the work of Viola and Jones [59] and Lienhart and Maydt [60], is used to detect the position of the

Figure 2.5: Schematic of the algorithm employed by Poh *et al.*

face. For each region detected as face, the algorithm returns the x and y coordinates along with the height and width of a square that describes the position of the face. The ROI is defined as the rectangle centered on the square found by the algorithm, with 60% of its width and 100% of its height. Whenever the algorithm is not able to find the ROI for a given frame, the ROI from the previous frame is employed. The average value of the red, green and blue channel of the pixels inside the ROI is stored on the signals $x_r[i]$, $x_g[i]$ and $x_b[i]$, respectively.
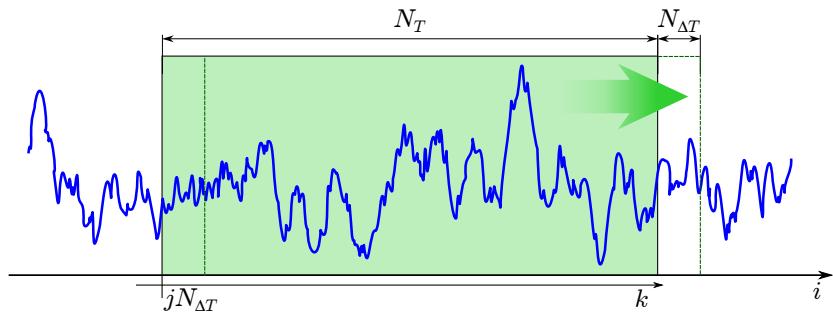
Figure 2.6: Signal windowing.

The normalization phase is executed in a window of duration T ($T = 30s$ in their algorithm), as shown in Figure 2.6. The window moves with ΔT seconds of increment ($\Delta T = 1s$). The normalized traces are computed removing the mean (DC component) from $x_c[i]$ (where c denotes one the three color channels) and adjusting its amplitude in order to obtain a signal with unitary variance, as follows:

$$y_c[j, k] = \frac{x_c[jN_{\Delta T} + k] - \mu_c[j]}{\sigma_c[j]}, \quad 0 \leq k < N_T. \tag{2.14}$$

The integer $j \geq 0$ is the window number and defines the window starting point. k is the relative position inside the window. We define N_T and $N_{\Delta T}$ as the number of frames comprised in T and ΔT seconds, respectively. The mean and variance (μ and σ, respectively) are given by:

$$\mu_c[j] = \frac{1}{M} \sum_{k=0}^{N_T - 1} x_c[jN_{\Delta T} + k] \tag{2.15}$$

and

$$\sigma_c[j]^2 = \frac{1}{M} \sum_{k=0}^{N_T - 1} \left(x_c[jN_{\Delta T} + k] - \mu_c[j]\right)^2. \tag{2.16}$$

For a fixed j, $y_c[j, k]$ is a vector of duration T over which we try to determine the subject HR. As j progresses, the window moves ΔT seconds along the signal, with a overlap of $(T - \Delta T)/T$.

Independent Component Analysis, based on the work of Cardoso [61], is used to separate the PPG signal from other noise components that are comprised in the $y_c[j, k]$ traces. This algorithm uses fouth-order cumulant tensors to automatically define weights for a linear mixture of the three channels in order to provide the best SNR, resulting in $z[j, k]$.

Finally, the Discrete Fourier Transform (DFT) is applied over $z[j, k]$ to obtain its spectrum $Z[j, v]$ and a peak detector determines the component of highest power within the range between 45 to 240 Beats per Minute (BPM), that correspond to the HR they expect to find for an adult individual. To account for noise, if the absolute difference between the current estimated pulse rate is above 12 BPM from the last computed value, the algorithm reject the actual estimation and searches for the next highest power component that meet this constraint. If no frequency peaks meet this criteria, then the current pulse frequency is retained.

Some works also have shown that arterial oxygen saturation (SpO_2) [62] can also be contact-less estimated using cameras by means of the light reflected by the skin [63, 64]. However, they require a device capable of capturing light in the infrared spectrum and a controlled illumination in order to provide a good estimation since SpO_2 is computed measuring the amount of light absorbed by oxyhemoglobin and de-oxyhemoglobin.

2.3.3 Micromovements

Balakrishnan *et al.* [56] also proposed an approach to estimate HR using videos, but instead of trying to capture the PPG signal, they focus their attention on the ballistocardiograph (BCG) signal, that correspond to subtle movements of the body due to the heart beats.

The head is subject to movement in most axis and can be considered, for small amplitude movements, an inverted pendulum. As the blood enters and leaves the head, propelled by the heart, micro-movements appears.

The ROI is defined as being the head with the eyes removed (found with Viola Jones face detector). Head movements are estimated using feature tracking and Principal Component Analysis is used to give more robustness against noise.

This approach is successful for detecting the HR and can be employed even when no skin is visible on the video (when the subject is using a mask, for example). However, it suffers many others limitations because respiration, posture changes and voluntary or involuntary movements, that have higher amplitude, dominate the trajectory of the tracking points. This makes it harder for the tracker to perceive the BCG signal, negatively affecting the estimation.

3 HEART RATE ESTIMATION THROUGH FACE DETECTION

This method is based upon the work of Poh *et al.* [13, 14]. We contributed to improved the robustness to noise adding an adaptive and derivative filter to boost the energy of the PPG signal over noise. We also propose and employ another approach to mix the information from the red, green and blue channels, avoiding the use of ICA that adaptively determine the weight for each color channel, in an attempt to enhance the SNR.

In this manuscript, we refer to this proposed method as "video heart rate estimation through face detection" (HR-FD) because it employs the face of the person being monitored as region of interest, from where the PPG signal will be extracted.

We present our method in Section 3.1. Some preliminary results are given in Section 3.2.

3.1 METHOD

Figure 3.1 presents a block diagram of the processing employed to estimate the HR from the video in this approach. This is very similar to the schematic for Poh *et al.* given in Figure 2.5, except for those blocks in red that were added or significantly modified.

$I[i]$ represents the i-th frame, that was acquired with a constant sampling rate, and $p[j]$ is the j-th HR detected. The signals x_r, x_g and x_b are computed from the ROI and normalized to the signals y_r, y_g and y_b following the same strategy, with a moving window with $T = 30s$ and $\Delta T = 0.5s$. The HR is calculate at each $0.5s$ and need a sequence of frames to be estimated.
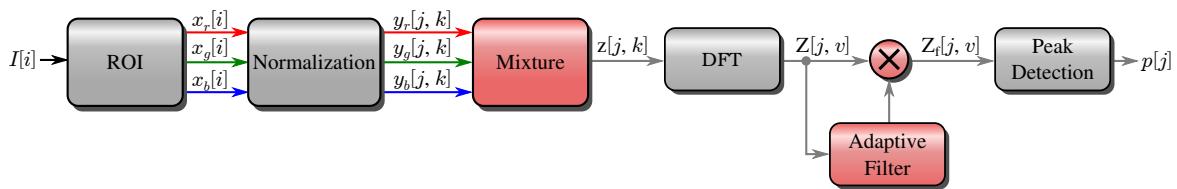
Figure 3.1: Schematic of the signal processing by HR-FD.

For the mixture stage, on the other hand, where y_r, y_g and y_b are linearly combined to provide a better SNR for the PPG signal, we removed the ICA to blindly separated the sources from the signal processing. In the ICA approach, three constants, α_r, α_g and α_b, are estimated, used to mix the signals as:

$$z[j,k] = \alpha_r y_r[j,k] + \alpha_g y_g[j,k] + \alpha_b y_b[j,k]. \tag{3.1}$$

However, we noticed in our work that even though the α values varies greatly from one video to another, or even from one signal site to another, their ratios are relatively constant, being almost independent from skin tone and scene illumination. Therefore, we normalized the α values, making them comparable between each other, using the constraints $\alpha_g > 0$ and $|\alpha_r|+|\alpha_g|+|\alpha_b| = 1$. Figure 3.2 presents the distribution obtained for these values in our database, where we can observe their tendency to concentrate in a given region.
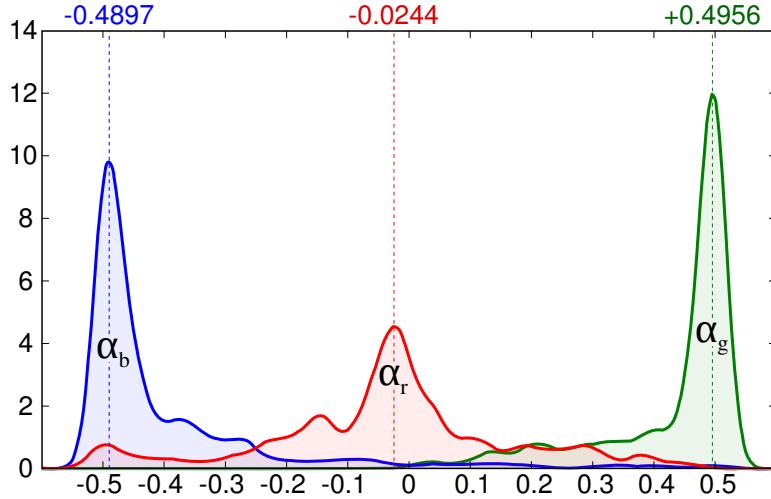
Figure 3.2: Distribution of the α_r, α_g and α_b values estimated using ICA, after normalization, in our database. The α values are represented by their corresponding colors

Due to this tendency, we decided to assume these values as constants given by their position of highest concentration, reducing the algorithm complexity as no blind source separation, which is computationally expensive, must be performed. Also, when the SNR is low or the noise characteristics are similar to the PPG signal, the ICA is not capable to precisely estimate the weights to employ in order to obtain a good SNR. This can leads to errors and instabilities on the estimation. As in our algorithm the weight of each channel is a constant, we are not susceptible to such errors.

The values employed are $\alpha_r = -0.0244$, $\alpha_g = +0.4956$ and $\alpha_b = -0.4897$, based on Figure 3.2. These values are in accordance with the literature because the green channel is the one presenting the highest energy for the PPG signal [8, 15]. But they are dependent on the camera parameters, such as sensor sensitivity, color filter, color and gamma correction. Thus, they must be determined for each camera.

After the mixture, for a fixed j, the signal $z[j, k]$ is zero padded to contain a total of $N_{FT} = 2^{14}$ elements. The magnitude of its DFT is computed, resulting in $Z[j, v]$, where v represents the frequency. We retain only those frequencies in the range going from 30 to 240 BPM that correspond to the values that we consider acceptable for a HR measurement for an adult individual. That is,

$$30\,\frac{N_{FT}}{f_s} \leq v \leq 240\,\frac{N_{FT}}{f_s}, \tag{3.2}$$

where f_s is the signal sampling frequency, given in frames per second.

An adaptive filtering is then applied over $Z[j, v]$, multiplying it by a mask $M[j, v]$, resulting in the signal

$$Z_f[j, v] = Z[j, v]M[j, v]. \tag{3.3}$$

The mask aim is to amplify $Z[j, v]$ for those frequencies that have a higher probability of be the HR frequency and attenuate others, reducing the effect of noise in the estimation. The mask is defined supposing that the HR varies slowly with time and that within $\Delta T = 0.5s$ it should be almost the same. Therefore, the previous signals, $Z[j - 1, v]$ and $Z[j - 2, v]$, provide a good estimation of where the HR peak should be.

In order to accommodate small variations of the HR, we first apply a convolution with a low-pass filter to $Z[j - 1, v]$ and $Z[j - 2, v]$ (the convolution is made in the frequency domain). This filter will horizontally stretch the peaks and is almost equivalent to apply an apodization function (or window function) [65] on the time domain. For the low-pass filter we chose $T_2[v]$, a triangular function with a bandwidth of 2 BPM, as shown in in Figure 3.3, that roughly allow a ± 1 BPM variation.

As $Z[j-2, v]$ is further away in time to the j-th signal than $Z[j-1, v]$, we apply again the same low-pass filter to it to accommodate highers HR variations. The mask is given by a combination of them as

$$M[j, v] = (T_2[v] * Z[j - 1, v]) \ (T_2[v] * T_2[v] * Z[j - 2, v]), \tag{3.4}$$

where $*$ denotes convolution.
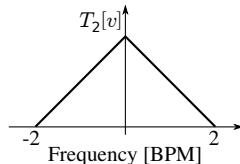
Figure 3.3: Low-pass filter employed to compute the adaptive filter mask.

Also, it was depicted by Xu *et. al.* [9] that use the derivative of the traces improve the HR detection performance as the noise tends to be more intense for low frequencies. Compute the derivative of a signal is equivalent to make it pass by a high-pass filter that amplify the signal proportional to its frequency. Hence, we decided to include a high-pass filter on the definition of the mask. The filter employed is shown in Figure 3.4. It apply a gain of 0.2 for frequencies inferior to 20 BPM and a gain of 1 for frequencies superior to 150 BPM. Between 20 and 150 BPM it applies a gain that varies linearly with frequency. This filter has a behavior similar to a derivative filter, but we restricted its actuation between 20 and 150 BPM to avoid over-attenuation of low frequencies, that could destroy useful information, and over-amplification of high frequencies, that could boost high frequency noise.
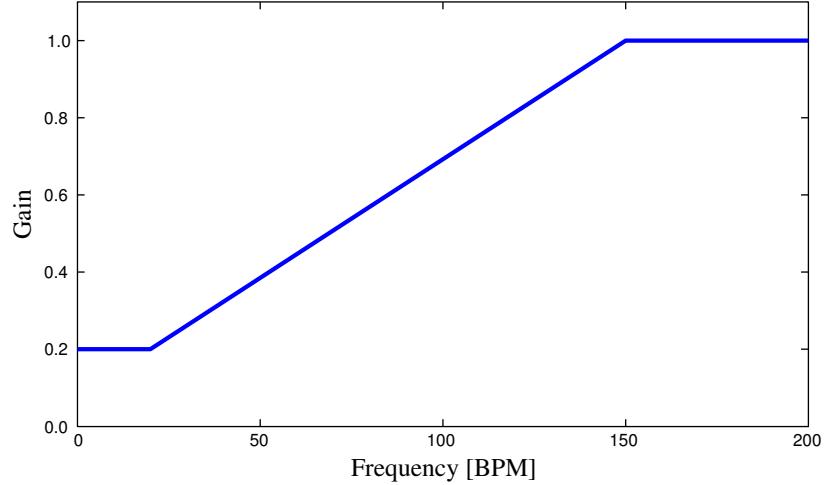
Figure 3.4: Derivative filter.

For simplification, we refer to this filter as derivative filter ($F_d[v]$). It is applied in cooperation with the adaptive filter to further improve the mask, that is now expressed as:

$$M[j,v] = F_d[v] \, (T_2[v] * Z[j-1,v]) \, (T_2[v] * T_2[v] * Z[j-2,v]) . \tag{3.5}$$

Figure 3.5 shows the application of the mask on a real signal. Note that the HR peak becomes more prominent on the filtered signal. This can be explained by two effects: (i) the peaks corresponding to the HR usually have a high amplitude and their amplitude and frequency changes slowly with time; (ii) those peaks corresponding to noise usually have a lower amplitude and their amplitude and frequency change faster than the HR with time. Hence, for regions where the signal is composed by noise the mask will apply an attenuation.
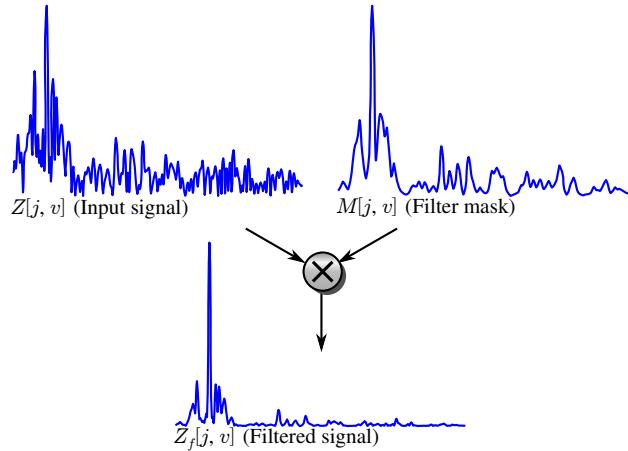
Figure 3.5: Adaptive filtering. The signal is multiplied by a mask in an attempt to attenuate noise and make the HR peak more prominent

If the assumptions made for the noise are not valid, that is, its frequency and amplitude do not vary with time, the amplitude of the mask on the corresponding frequency will be proportional the noise peak amplitude. Thus, the multiplication by the mask will have an effect similar to

square the input signal at that frequency. As squaring a signal do not alter the ordering of the peaks amplitude, the signal filtered by means of multiplication by this mask will have no negative effect, in terms of estimation, for noise components that varies slowly with time and will attenuate the noise components that variate faster.

Finally, from the filtered signal, the position of the peak of higher amplitude is found and its frequency is the j-th estimated HR frequency for the subject ($p[j]$). If the absolute difference between $p[j]$ and $p[j-1]$ is higher than 12 BPM, the algorithm search between the four highest peaks which one is the nearest to $p[j-1]$ with an absolute difference inferior to 12 BPM. If none of them meet this constraint, the frequency of the peak of higher amplitude is used as estimation.

3.2 PRELIMINARY RESULTS

Figure 3.6 presents the HR estimated with the proposed method in comparison with that of Poh [13] for three different videos. For the ground truth we used an finger oximeter to monitor the subject HR at the same time that the video was captured.

In the first two videos, I and II, the subject were asked to remain still, moving as little as possible. In video III, the subject was talking and moving freely while the video was captured. It can be seem that for the video I the performance of our method and that of Poh where very similar. Indeed, they outcome the same values and their curves are overlapped. For video II, between 20 and 33s, the algorithm of Poh diverged from the HR captured from the oximeter while our approach remained very similar to it. We attribute this gain in performance due the adaptive filtering employed that improves the SNR (see Figure 3.7).

In the third video, on the other hand, both algorithms diverged from the values measured from the oximeter due to the artifacts introduced from movement.

The effect of the adaptive filtering can be observed in Figure 3.7, that shows the signal used to feed the peak detector in both algorithms, $Z[j,v]$ for Poh and $Z_f[j,v]$ for our method. The filter was able to improve the SNR on all instants, except for the last video, as it can be seen. For video I, the SNR was already high before the filtering, therefore both algorithms were able to correctly estimate the HR. The same is valid for video II, as it can be see in (b), except for the interval between 20 to 33s, where Poh algorithm failed due to noise. As the adaptive filtering removed most of this noise, the performance of our algorithm was little influenced by it.

However, video III shows that both algorithms are not robust to movements of higher amplitude and further improving is necessary concerning this case.
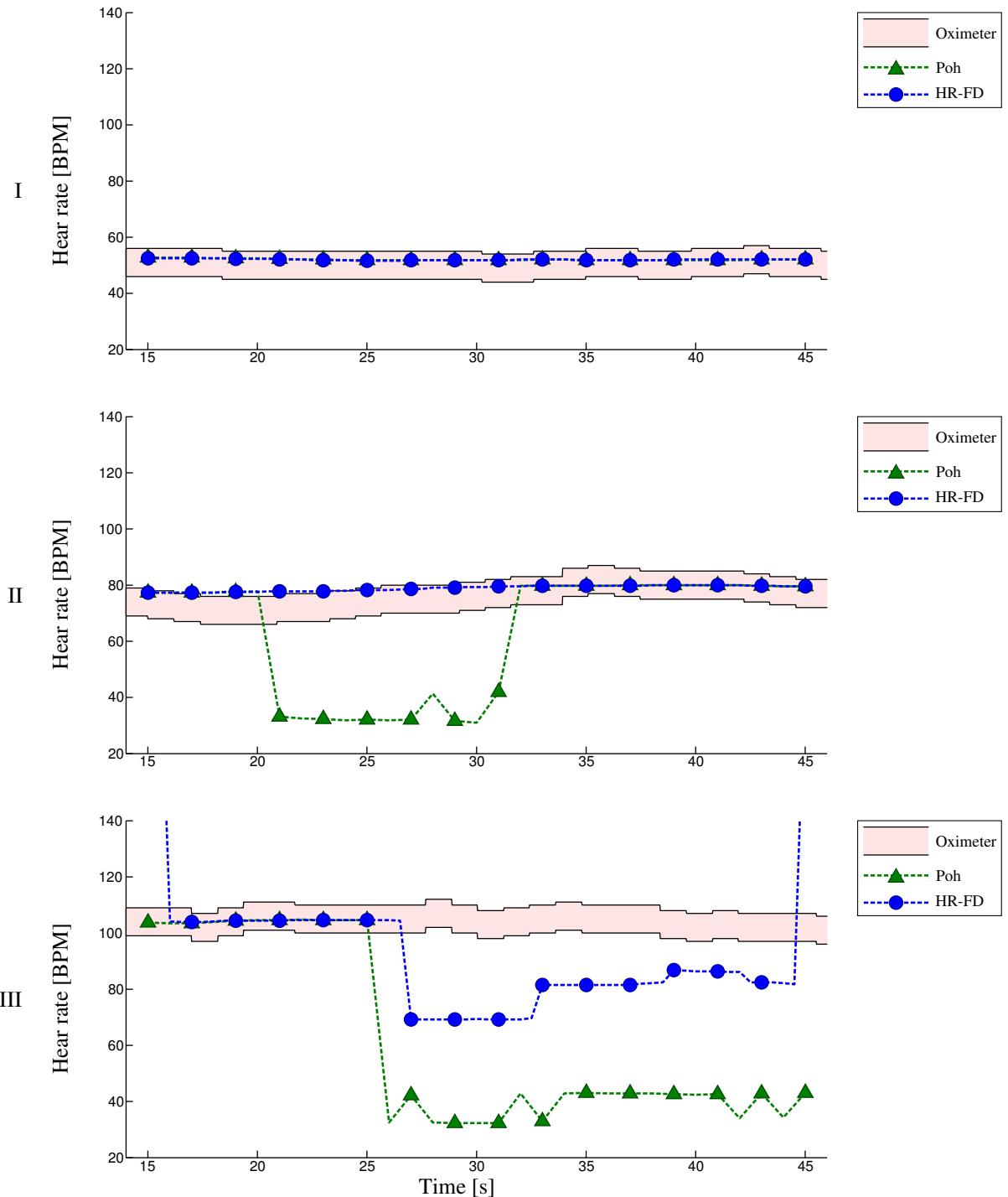
Figure 3.6: Performance of the HR detection for three different videos. In I and II we asked the subject to stay as still as possible and in III the subject is freely talking and moving as the video is captured.
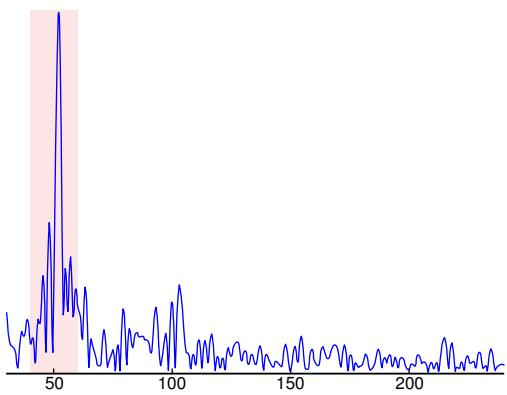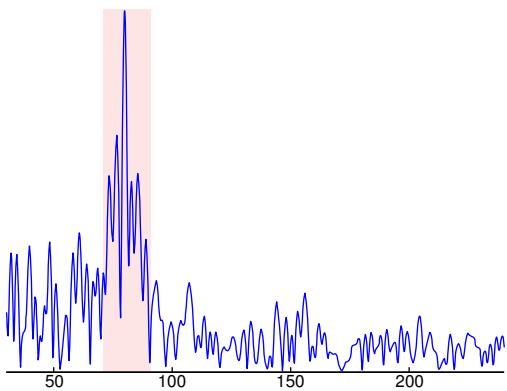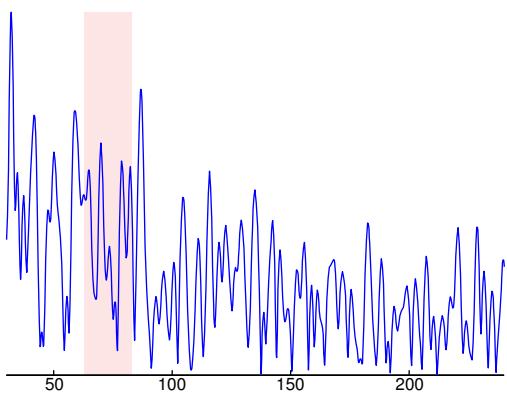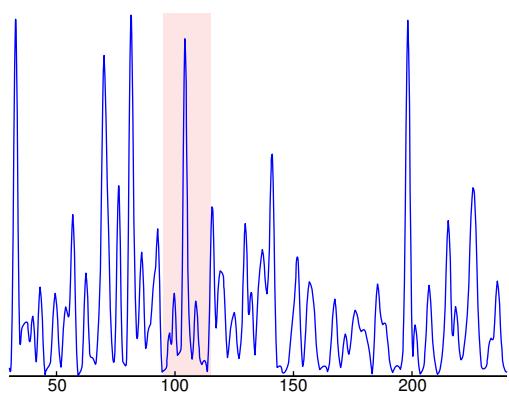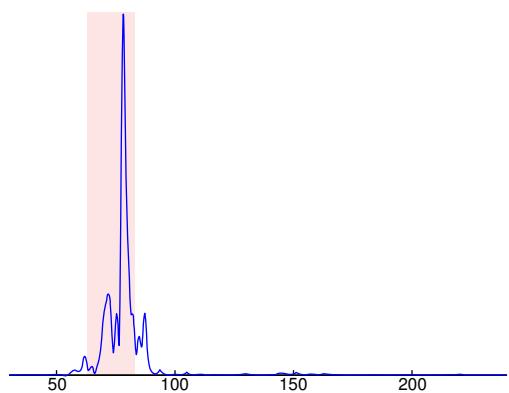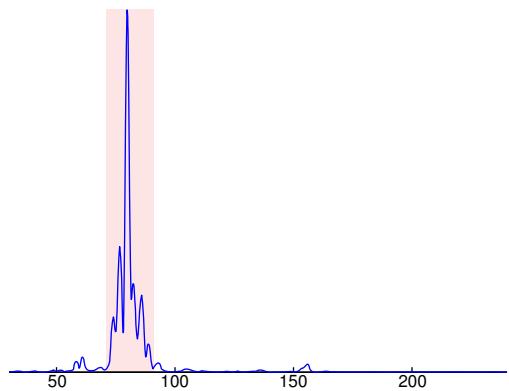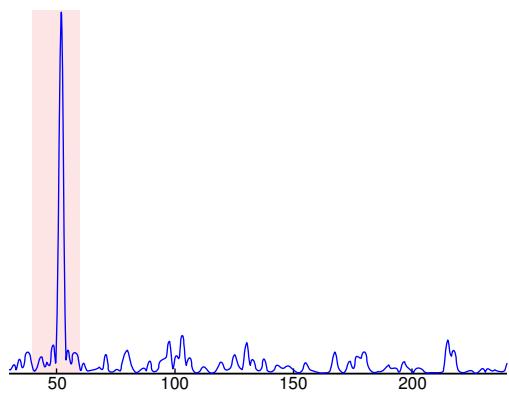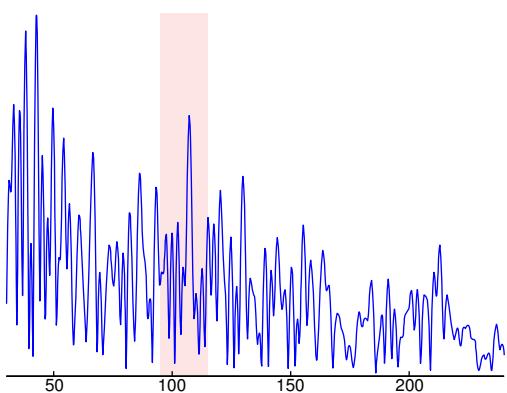
Figure 3.7: Signal fed to the peak detector as processed by the algorithm of Poh *et al.* and the proposed method. These signals were extracted from the same videos used in Figure 3.6: a) I at 22s; b) II at 38s; c) II at 25s and d) III at 34s. The colored rectangle correspond to the oximeter reading at that time.

4 HEART RATE ESTIMATION THROUGH MICRO-REGION TRACKING

This method, denominated "video heart rate estimation trough micro-region tracking" (HR-MRT), emerged as a modification of the HR-FD method in an attempt to produce an algorithm more robust to movements, avoiding the problem found in Figure 3.6-III.

Movement affects the ROI. The face deforms while someone speaks and the movement of the lips, eyelid, eyebrows, etc, are all captured by the values of x_r, x_g and x_b (the average value for red, green and blue inside the ROI) and will introduce artifacts that makes the task of HR estimation more difficult.

If this artifacts are of small amplitude or their frequency are different than the range of frequencies where we search for the HR, they will have little influence on the estimation. That is the case for videos I and II in Figure 3.6, where most of the artifacts are due to blinking and other small movements, but not for video III, where the movements have higher amplitudes.

Therefore, we modify the ROI to make it more robust to movements. Figure 4.1 is a schematic of how the ROI is defined in this method. We divide the video in blocks with the same duration, segment the first frame of each block in micro-regions and extract some features from them to track. The tracking of these features enable us to estimate the parameters of an affine transformation that describe how the micro-regions move with time. We then extract the red, green and blue traces for each micro-region, normalize and mixture them with the same procedure employed for HR-FD. From the DFT for each micro-region we feed a clustering algorithm that will define the ROI as a combination of micro-regions where the PPG signal is visible. The output of the clustering algorithm are the red, green and blue traces. The subsequent steps to estimate the HR were omitted since they are the same employed for HR-FD, displayed in Figure 3.1.
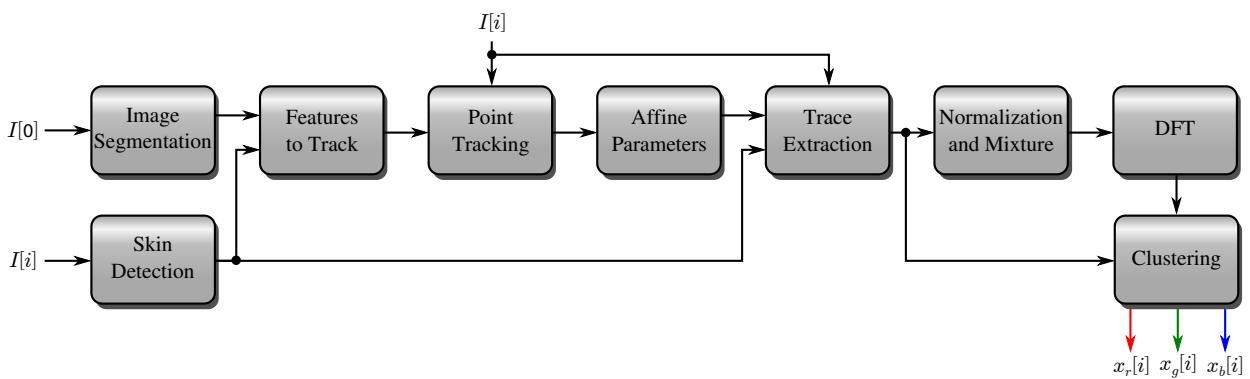
Figure 4.1: Block diagram for the ROI definition on HR-MRT.

Section 4.1 describes the skin detection algorithm employed to ignore non-skin pixels from the micro-regions, Section 4.2 shows how we divide the image in micro-regions to take movements

into account and Section 4.3 how we use tracking to compensate for movements. Section 4.4 explain the clustering algorithm used to define the ROI.

4.1 SKIN DETECTION

As the PPG signal is only present in pixels that correspond to the skin of the person being monitored, a skin detector is a good tool to eliminate non-skin pixels that could add noise to the red, green and blue traces computed on the ROI. Hence, we decided we employ a skin detector to ignore non-skin pixels following three strategies. In the first strategy we created a look-up table using the histogram based approach described in Section 2.2.3. We used the database provided by Jones and Rehg [44][1] that is composed of images obtained from the internet for which we know the ground truth. The database is comprised by 3789 images containing skin from people in different backgrounds and 6187 images where no skin is present. The color space employed is the YCbCr as it is the preferred among researchers in the area. We do not drop the luminance component and the histograms have 256^3 bins. We used a cubic filter of size 3^3 to smoth the histogram before creating the look-up table; .

As it can be seen in Figure 4.2-(b), this algorithm results in a great quantity of false positives because, in real life images, the clusters for skin and non-skin pixels overlap. Therefore, in the second strategy we combine this skin detector with the Viola-Jones face detector to eliminate those pixels that do not correspond to the face. Only pixels in the rectangle with 100% of the height and 80% of the width of the region found by the face detector are kept.

| a) Original | b) Lookup table | c) Face detect | d) Manual |
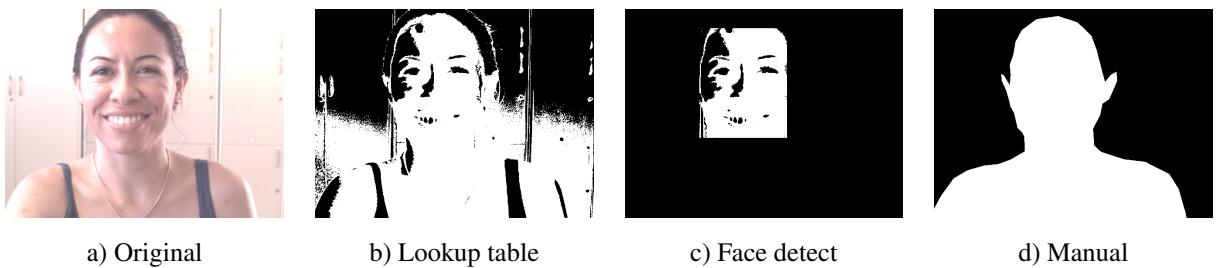
Figure 4.2: The image in a) was provided to the skin detection algorithm using b) histogram based approach employing a lookup table, c) the histogram approach combined with Viola-Jones face detector and d) manual detection.

In the third strategy we manually selected the skin pixels to observe the performance of our algorithm when we eliminate its dependence on the performance of the skin detector.

---

[1]https://drive.google.com/folderview?id=0Bz-X0E2bqx9YcW9GaEM0OS0xb28&usp=sharing

4.2 MICRO-REGIONS

The micro-regions are defined in the first frame of each block of the video and are segments with uniform color, found by a segmentation algorithm. We used the watershed method, an idea introduced by Beucher and Lantuéjoul [66] based in a topological representation of an image, to segment the frame. This method was chosen because of its simplicity and also because it divides the image in several segments and we can indirectly control the size and number of segments.

Prior to the segmentation with watershed we usually compute the gradient modulus of the image. The gradient is more intense for regions of transition, like the edges between one object to another, and less intense elsewhere. It is a good representation of the image borders. The regions of high gradient can be seen as mountains separating valleys (low gradient).

Referring to Figure 4.3, if we let a drop of water fall in a given position, it will flow down the mountains formed by the gradient until it arise to a local minimum. As more water drops arrive they start to form a puddle. We can intuitively interpret the watershed segmentation by means of these drops. Those pixels that flow to the same puddle belong to the same region (they are in the same catchment basin of that minimum) and the gradient is used as a base to separate one region from another.
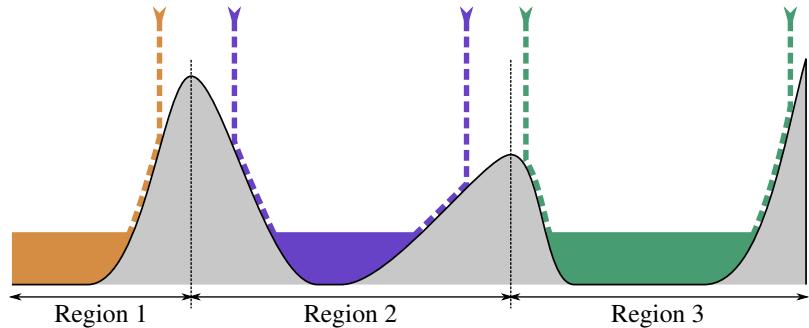
Figure 4.3: Two dimensional representation of the watershed segmentation method. A region correspond to all positions for which a drop of water, flowing down the hills, would fall in the same puddle.

The algorithm to perform the watershed segmentation can be given in three steps:

1. Create a group containing all non-labeled pixels of the image. As initially none of them is labeled, this group will contain the entire image;

2. From the group of non-labeled pixels, extract those of minimal altitude and attribute to them the label of an adjacent labeled pixel. If there is no adjacent labeled pixel, atribute a new label to it.

3. Repeat step 2 until there is no more non-labeled pixels left.

One disadvantage of the watershed method is its tendency to over-segmentation. Thus, it is a standard practice to smooth the image before applying it, eliminating or attenuating gradients

of low amplitude in textured portions of the image. In our work we employed a bilateral filter, a non-linear edge preserving low-pass filter that was first proposed by Tomasi and Manduch [67]. This filter smooths a pixel by averaging them with neighboring pixels, similar to isotropic filters, but add a penalty based on their color difference. Let $I(x, y)$ be the original image that we want to filter, the filtered image is given by

$$I_{BF}(x, y) = \sum_{(i,j) \in \Theta} I(x + i, y + j) \, D_{ij} \, P\{\Delta I_{ij}(x, y)\}, \qquad (4.1)$$

$$\Delta I_{ij}(x, y) = I(x + i, y + j) - I(x, y),$$

where Θ defines the neighborhood. D_{ij} is a distance penalty function that reduces the weight of a pixel based on how far it is from the pixel being filtered and $P\{\Delta I_{ij}(x, y)\}$ introduces a penalty based on how different the pixels are. We suppose that pixels that are in different sides of the border will have a significant difference in color and those in the same side will be rather similar. Therefore, the second penalty try to adjust the filtering in order to take the borders in account. These functions are subject to the constraint

$$\sum_{(i,j) \in \Theta} D_{ij} \, P\{\Delta I_{ij}(x, y)\} = 1. \qquad (4.2)$$

For our work we used a Gaussian function to express these penalties in a rectangular neighborhood. The filtering is then expressed as

$$I_{BF}(x, y) = \left(\sum_{i=-N}^{N} \sum_{j=-N}^{N} I(x + i, y + j) W_{ij}(x, y) \right) \Bigg/ \left(\sum_{i=-N}^{N} \sum_{j=-N}^{N} W_{ij}(x, y) \right), \qquad (4.3)$$

where, in this case, N defines the size of the neighborhood and

$$W_{ij}(x, y) = exp\left(-\frac{i^2 + j^2}{2\sigma_1^2} \right) exp\left(-\frac{|\Delta I_{ij}(x, y)|^2}{2\sigma_2^2} \right), \qquad (4.4)$$

is the weight attribute to pixel at position $(x + i, y + j)$ to compose the filtered pixel at (x, y).

Equation (4.4) has two components. The first one is the distance penalty. The second term introduces the non-linearity and anisotropy on the filtering trying to take in account the borders.

Figure 4.4 shows the application of this filter in the image of Lena with noise added. The filter is capable of reducing the noise without corrupting the borders. However, one disadvantage of this algorithm is its susceptibility to create a carton-like effect on the filtered image.

From the filtered image we compute its luminance, $Y(x, y)$, the same way as the Y component of the YCbCr color space (see Section 2.2.1.3). The gradient is given by the horizontal and vertical

a) Original   b) Noisy (SNR = 16.3 dB)   c) Filtered (SNR = 23.6 dB)   d) Section

Figure 4.4: Image smoothing with bilateral filtering: a) is the original image with a dimension of 512x512 pixels and in b) a Gaussian noise was added to the image in a). c) shows the result of the filtering with $\sigma_1 = 3$, $\sigma_2 = 0.15$ and $N = 8$. Figure d) is a section of Lena's arm showing in close up the before and after filtering.

derivative of the luminance. To compute the horizontal and vertical derivative we use the kernels

$$K_h = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{ and } K_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix},$$ (4.5)

respectively, and the squared gradient is given by

$$G(x,y)^2 = (Y(x,y) * K_h)^2 + (Y(x,y) * K_v)^2.$$ (4.6)

We further smooth $G(x,y)^2$ with a rectangular kernel to attenuate the gradient on textured regions and avoid over segmentation and we apply the watershed method to segment the image (see Figure 4.5).

Finally, after segmentation, we ignore those micro-regions that contain less than 80% of skin pixels.

4.3 MICRO-REGION TRACKING

Each micro-region is delimited by the pixels in its border. Hence, if we can track how this pixels position evolved with time we can describe how the micro-region moved and deformed in the video sequence. In this fashion, we would be correcting the artifacts introduced by movement in the video. Although, tracking the pixels on the borders of the micro-regions is not a easy task because some of them may be in low textured regions where tracking algorithms fail to precisely estimate the pixel disparity and the amount of pixels to track would slow down the algorithm.

Therefore, for each micro-region segmented, we select a set of easy to track points that are inside or on the border of the micro-region (up to 12 points for each micro-region). We suppose that the pixels inside the region are subject to the same movement as the border pixels. Thus,

a) Original image

b) 9x9   c) 15x15   d) 25x25
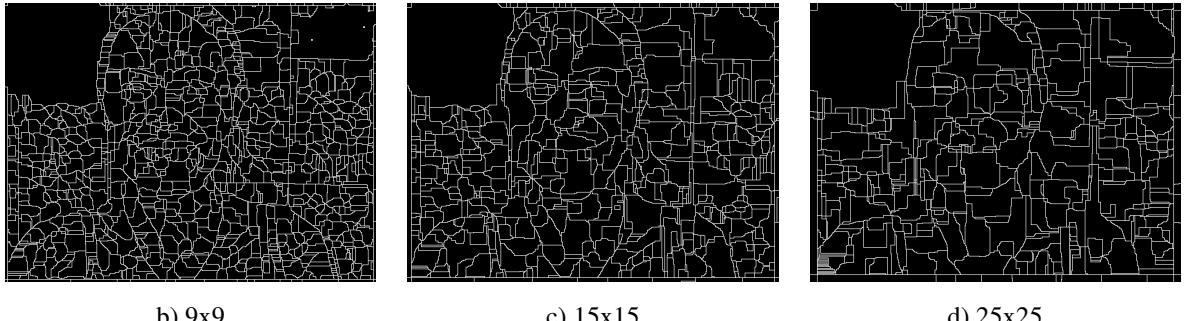
Figure 4.5: Segmentation of the image in a) using watershed with different kernel sizes to smooth the gradient. The bilateral filtering was performed using $\sigma_1 = 3$, $\sigma_2 = 0.06$ and $N = 5$.

we use these points to estimate how the micro-region moved and deformed with time. The set of points is selected using the algorithm of Shi and Tomasi [68] as implemented in OpenCV 2.4. This algorithms finds the most prominent corners within the micro-region. This set of points are then tracked with the Lucas-Kanade algorithm as implemented by Yves [69].

The optical flow computed with Lucas-Kanade method may contain some noise. This noise should not affect short sequences of video, but for longer sequences it start to add a drift to the motion estimation that can lead to incorrect values for the optical flow, which can spoil the micro-region tracking.

To overcome this issue, one can filter the data to attenuate this noise in order to reduce its effect for longer sequences. But before that we need to define a model to the data to decide how to filter it. In this work we suppose that, for a short interval, the movement of the objects in the scene can be described by a polynomial function of a given order. Let $\mathbf{p}(t) = [x_t(t), y_t(t)]^T$ ($\bullet^T$ means transpose) be the position of a pixel that we are tracking in time t, then we can express $\mathbf{p}(t)$ as

$$\mathbf{p}(t) = \sum_{n=0}^{N_{poly}} \mathbf{r}_n \frac{t^n}{n!}, \tag{4.7}$$

where $\mathbf{r}_n = [x_n, y_n]^T$ are constants and N_{poly} is the polynomial order. The higher the polynomial order the better it can represent complex movements, but it becomes more sensible to noise. A good trade-off between data representation and noise cancellation is found for $N_{poly} = 3$. Thus,

for simplicity of reading, we will consider only the case of third order polynomials models in this chapter. Results for other orders can be found following the same steps. A demonstration of this statement will be presented in the next chapter. Therefore we have

$$\mathbf{p}(t) = \begin{bmatrix} x_t(t) \\ y_t(t) \end{bmatrix} = \begin{bmatrix} x_0^n \\ y_0^n \end{bmatrix} + \begin{bmatrix} v_x^n \\ v_y^n \end{bmatrix} t + \begin{bmatrix} a_x \\ a_y \end{bmatrix} \frac{t^2}{2} + \begin{bmatrix} u_x \\ u_y \end{bmatrix} \frac{t^3}{6}, \qquad (4.8)$$

or, in a more compact way,

$$\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{v}t + \mathbf{a}\frac{t^2}{2} + \mathbf{u}\frac{t^3}{6}, \qquad (4.9)$$

where $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$ and $\mathbf{u}$ are constants vectors that define the initial position, velocity and acceleration.

Let us now assume that we have two reference images, R_1 and R_0 that are consecutive in the video sequence (R_1 being first) for which we know the position of the point we are tracking, $\mathbf{p}^1$ and $\mathbf{p}^0$, respectively. We also have six other images on the sequence, I_1, I_2, I_3, I_4, I_5 and I_6, for which we don't know the tracking point position yet. Their position will be estimated using Lucas-Kanade method.

The sequence of images, $[R_1, R_0, I_1, I_2, I_3, I_4, I_5, I_6]$, form a sub-sequence of the video, meaning that they are all consecutive. We suppose that Equation (4.9) is a good model to represent the movement of the tracking pixel in this sub-sequence between frame R_0 and I_6.

Using the reference frames we estimate the optical flow as schematized by Figure 4.6. We denote as $\mathbf{p}_n^0 = [x_n^0, y_n^0]^T$ the position estimated for frame I_n using information coming from reference R_0 and as $\mathbf{p}_n^1$ when coming from reference R_1. From this estimated values we find the constants in Equation (4.9) that minimizes the quadratic error. We assume that R_1 and R_0 correspond to $t = -1$ and 0, respectively, and I_n correspond to $t = n$.
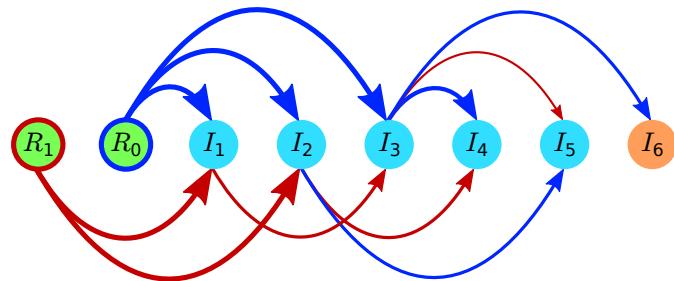
Figure 4.6: Optical flow estimation scheme. The arrows indicate from which to which frame the optical flow is estimated using the Lucas-Kanade method. Blue arrows represent that the flow was estimated using information coming from reference R_0 and red arrows from reference R_1.

The constants can be found by linear algebra, solving the equation

$$
\begin{bmatrix}
1 & 0 & 0^2/2 & 0^3/6 \\
1 & 1 & 1^2/2 & 1^3/6 \\
1 & 1 & 1^2/2 & 1^3/6 \\
1 & 2 & 2^2/2 & 2^3/6 \\
1 & 2 & 2^2/2 & 2^3/6 \\
1 & 3 & 3^2/2 & 3^3/6 \\
1 & 3 & 3^2/2 & 3^3/6 \\
1 & 4 & 4^2/2 & 4^3/6 \\
1 & 4 & 4^2/2 & 4^3/6 \\
1 & 5 & 5^2/2 & 5^3/6 \\
1 & 5 & 5^2/2 & 5^3/6 \\
1 & 6 & 6^2/2 & 6^3/6
\end{bmatrix}
\begin{bmatrix}
x_0 & y_0 \\
v_x & v_y \\
a_x & a_y \\
u_x & u_y
\end{bmatrix}
=
\begin{bmatrix}
x^0 & y^0 \\
x_1^0 & y_1^0 \\
x_1^1 & y_1^1 \\
x_2^0 & y_2^0 \\
x_2^1 & y_2^1 \\
x_3^0 & y_3^0 \\
x_3^1 & y_3^1 \\
x_4^0 & y_4^0 \\
x_4^1 & y_4^1 \\
x_5^0 & y_5^0 \\
x_5^1 & y_5^1 \\
x_6^0 & y_6^0
\end{bmatrix}
\tag{4.10}
$$

that can be more compactly written as

$$
TA = P, \tag{4.11}
$$

where T is the time matrix, A the parameters matrix and P the position matrix.

Equation (4.11) falls in the category of a very well known mathematical problem: that of linear least squares fitting for an overdetermined system of linear equations [70]. These problems are convex and have a closed-form solution that is unique and given by

$$
A = (T^T T)^{-1} T^T P \tag{4.12}
$$

Now that we know the parameters we can recalculate the position of the tracking pixel in the frame I_n, $1 \leq n < 6$ resulting in

$$
T'(T^T T)^{-1} T^T P = FP, \tag{4.13}
$$

$$
T' =
\begin{bmatrix}
1 & 1 & 1^2/2 & 1^3/6 \\
1 & 2 & 2^2/2 & 2^3/6 \\
1 & 3 & 3^2/2 & 3^3/6 \\
1 & 4 & 4^2/2 & 4^3/6 \\
1 & 5 & 5^2/2 & 5^3/6
\end{bmatrix}. \tag{4.14}
$$

$F = T'(T^T T)^{-1} T^T$ is therefore the filtering matrix that will eliminate from the set of points P the movements that do not obey the model given by Equation (4.9), represented in Figure 4.7, from where we can see that it corresponds to a low-pass filter, as expected. A higher (or lower) order model could also be employed, but as the order increases it starts to become unable to attenuate noise as they can be represented by the high order components of the model.
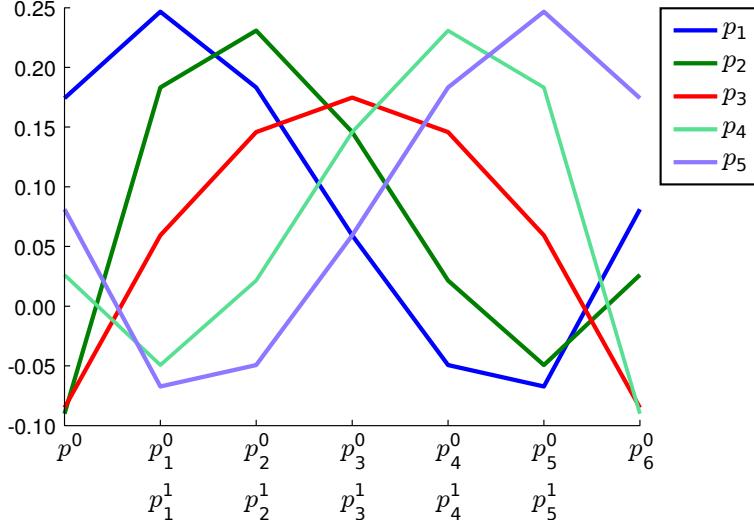
Figure 4.7: Feature tracking filtering. Shows the value of the filtering matrix F that multiply each component of the position matrix P.

Frame I_6 was used only to calculate the filtering parameters, but the filtered pixel position is not computed for this frame. We advance in the video sequence. I_4 and I_5 become now the reference and we use the same approach to estimate the optical flow for I_6, I_7, I_8, I_9, I_{10} and I_{11}. We keep advancing until all the video sequence is covered.

Finally, after tracking how the set of points for a given micro-region evolved with time using the above algorithm, we can determine how the micro-region moved. However, to this point, we only know how a set of points inside the region evolved with time. To use the information of the set of tracking points we suppose that the k-th points in the border of the micro-region, given by $\mathbf{b}^k(t) = [x_b^k(t), y_b^k(t)]^T$ in time t, undergo an affine transformation of the form

$$\begin{bmatrix} x_b^k(t) \\ y_b^k(t) \end{bmatrix} = R(t) \begin{bmatrix} x_b^k(0) \\ y_b^k(0) \end{bmatrix} + \begin{bmatrix} d_x(t) \\ d_x(t) \end{bmatrix}, \tag{4.15}$$

where R is a 2x2 affine matrix transformation and d_x and d_y are the translations. We don't know these parameters, but we can estimate them using the set of tracking points with the assumption that the tracking points undergo the same transformation as the border points. Let $[x_t^n(t), y_t^n(t)]^T$ be the n-th tracking points in time t for the micro-region we are processing, then

$$\begin{bmatrix} x_t^n(t) \\ y_t^n(t) \end{bmatrix} = R(t) \begin{bmatrix} x_t^n(0) \\ y_t^n(0) \end{bmatrix} + \begin{bmatrix} d_x(t) \\ d_x(t) \end{bmatrix}, \tag{4.16}$$

We can eliminate the translation in Equation (4.16) by subtracting each side of the equation by the mean value of the tracking point position in time, $[\overline{x}_t(t), \overline{y}_t(t)]^T$,

$$\begin{bmatrix} x_t^n(t) \\ y_t^n(t) \end{bmatrix} - \begin{bmatrix} \overline{x}_t(t) \\ \overline{y}_t(t) \end{bmatrix} = R(t) \left(\begin{bmatrix} x_t^n(0) \\ y_t^n(0) \end{bmatrix} - \begin{bmatrix} \overline{x}_t(0) \\ \overline{y}_t(0) \end{bmatrix} \right) \tag{4.17}$$

32

Equation (4.17) can be divided in 5 simple transformations. Ignoring translation, for simplicity, we have:

$$\text{Scaling} \qquad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \alpha(t) \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$$

$$\text{Rotation} \qquad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} cos(\theta(t)) & -sin(\theta(t)) \\ sin(\theta(t)) & cos(\theta(t)) \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$$

$$\begin{matrix} \text{Vertical} \\ \text{Shearing} \end{matrix} \qquad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha_v(t) & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$$

$$\begin{matrix} \text{Horizontal} \\ \text{Shearing} \end{matrix} \qquad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1 & \alpha_h(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$$

$$\text{Mirroring} \qquad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} (-1)^{m(t)} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$$

α is the scaling factor (a positive value), θ the anticlockwise angle of rotation, α_h and α_v are the horizontal and vertical shearing and $m = \{0, 1\}$ is a integer that indicate if the points suffered or not a horizontal mirroring.

If we restrict the affine transformation to only rigid transformation, ignoring shearing and mirroring, matrix R can be expressed as

$$R = \alpha \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} = \begin{bmatrix} a_{11} & -a_{21} \\ a_{21} & a_{11} \end{bmatrix}. \qquad (4.18)$$

On the other hand, allowing shearing and mirroring,

$$R = \alpha \begin{bmatrix} (-1)^m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & \alpha_h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha_v & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad (4.19)$$

These two distinct transformation are referred, in this work, as rigid affine transform and full affine transform, respectively.

To solve for R we use the procedure described by Lawson and Hanson [70]. We find the parameters that minimize the squared error. The solution is given in terms of sums

$$C_{xx}(t) = \sum_n \left[x_t^n(0) - \overline{x}_t(0) \right] . \left[x_t^n(t) - \overline{x}_t(t) \right]$$

$$C_{xy}(t) = \sum_n \left[x_t^n(0) - \overline{x}_t(0) \right] . \left[y_t^n(t) - \overline{y}_t(t) \right]$$

$$C_{yx}(t) = \sum_n \left[y_t^n(0) - \overline{y}_t(0) \right] . \left[x_t^n(t) - \overline{x}_t(t) \right]$$

$$C_{yy}(t) = \sum_n \left[y_t^n(0) - \overline{y}_t(0) \right] . \left[y_t^n(t) - \overline{y}_t(t) \right]$$

$$D_{xx} = \sum_n \left[x_t^n(0) - \overline{x}_t(0) \right]^2$$

$$D_{xy} = \sum_n \left[x_t^n(0) - \overline{x}_t(0) \right] . \left[y_t^n(0) - \overline{y}_t(0) \right]$$

$$D_{yx} = \sum_n \left[y_t^n(0) - \overline{y}_t(0) \right] . \left[x_t^n(0) - \overline{x}_t(0) \right] = D_{xy}$$

$$D_{yy} = \sum_n \left[y_t^n(0) - \overline{y}_t(0) \right]^2$$

Then, for Equation (4.18) we have

$$R = \frac{1}{D_{xx} + D_{yy}} \left(\begin{bmatrix} C_{xx} & -C_{xy} \\ C_{xy} & C_{xx} \end{bmatrix} + \begin{bmatrix} C_{yy} & C_{yx} \\ -C_{yx} & C_{yy} \end{bmatrix} \right) \tag{4.20}$$

and for Equation (4.19)

$$R = \begin{bmatrix} C_{xx} & C_{yx} \\ C_{xy} & C_{yy} \end{bmatrix} \begin{bmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{bmatrix}^{-1}. \tag{4.21}$$

We ignored the time dependency for simplicity of writing, but the reader must notice that matrix R varies with time.

The translation can be found from Equation (4.17) as

$$\begin{bmatrix} d_x(t) \\ d_y(t) \end{bmatrix} = \begin{bmatrix} \overline{x}_t(t) \\ \overline{y}_t(t) \end{bmatrix} - R(t) \begin{bmatrix} \overline{x}_t(0) \\ \overline{y}_t(0) \end{bmatrix} \tag{4.22}$$

Applying this affine transformation to the border of the micro-region we can determine how it evolved with time from their initial position and where the micro-region is in time t. It can be shown[2] that if the micro-region has a total area of A_0 in $t = 0$ and undergo an affine transformation, its area in time t is given by $A_0.det(R(t))$. We use this function as a metric to ignore

---

[2]The demonstration was omitted here. The reader can refer to `http://www.mathopenref.com/coordpolygonarea.html` and `https://www.math.wisc.edu/~robbin/461dir/coordinateGeometry.pdf` for a rigorous mathematical demonstration

those micro-regions for which its area became bigger than twice or smaller than half its initial size at some point as it can indicate that there was a misleading during point tracking or affine transformation estimation. Finally, we computed the mean value of red, green and blue for each micro-region over time. We ignore those pixels that were not detected as skin-pixel from the traces computation. The traces are then normalized following the same procedure as for the HR-FD and we compute its DFT.

4.4 CLUSTERING

The traces extracted for each micro-region may contain different levels of noise. For example, when the skin detector do not provide a good segmentation, some micro-regions may be composed mainly by background pixels, where the PPG signal is not present. For the micro-regions composed mainly by true skin pixels, the energy of the PPG signal and the noise may vary due to the characteristics of the skin on the corresponding region, such as beard, vasculature, makeup, motion noise.

We want to average the traces found for the micro-regions in a single trace to be employed for HR estimation. To obtain a good SNR we try to ignore those micro-regions where the PPG signal is not visible or where the noise energy obscures it. The clustering algorithm is used to resolve which micro-regions to use in order to maximize the SNR. It groups in a cluster those micro-regions that present similar traces comparing their Fourier transform. Based on the assumption that most micro-regions contain the PPG signal, we select the cluster of micro-regions with the highest number of elements and ignore the remaining. In this fashion, the algorithm automatically selects the ROI.

Section 4.4.1 presents a new distance metric used to compute the similarity between the traces of micro-regions and Section 4.4.2 the clustering algorithm *per se*.

4.4.1 Distance Metric

For each micro-region, we mix their red, green and blue traces using the weights given in Section 3.1 and we compute its DFT. Those micro-regions corresponding to the skin of the individual being monitored will have a Fourier transform composed by the PPG signal plus an additive noise and those that do not correspond to the skin will be primarily composed by noise. Hence one can expect that the Fourier transform of some micro-regions will be similar between each other because they carry the same signal.

However, even though some micro-regions have similar Fourier transform, their amplitude can vary significantly from one to another as a result from shadows, changes in the characteristics of the skin, *e.g.* tone, texture, beard, blood circulation, etc. Therefore, traditional approaches to compare functions, such as Euclidean distance, are not suitable in our case. Thus the necessity to

create a new metric capable of comparing the similarity between Fourier transforms that is robust to the scale.

Let $F_1[v]$ and $F_2[v]$ be the magnitude of two discrete Fourier transform that we want to compare, where v is the discrete frequency. They are said to be similar if they have peaks at the same frequencies, with amplitudes proportional to each other *i.e.*, exist a real constant $k > 0$ which yields

$$F_2[v] \approx k.F_1[v], \ v_{min} \leq v \leq v_{max}. \tag{4.23}$$

Where v_{min} and v_{max} define the range of frequencies where the two Fourier transforms are compared. This range is chosen as those frequencies where we expect the HR to be, from 30 to 240 BPM. Otherwise, when Equation (4.23) is not a good approximation, the functions are considered to be not similar.

One simple way to find the value of the constant k for equation (4.23) is to normalize $F_1[v]$ and $F_2[v]$ by their root mean squared amplitude (RMS amplitude),

$$A_1 = \sqrt{\frac{1}{N} \sum_i F_1[v]^2} \ \ \text{and} \ \ A_2 = \sqrt{\frac{1}{N} \sum_i F_2[v]^2}, \tag{4.24}$$

respectively, where N is the number of elements of F_1 and F_2. Therefore, equation (4.23) becomes:

$$\frac{F_2[v]}{A_2} \approx \frac{F_1[v]}{A_1} \tag{4.25}$$

Nevertheless, the RMS amplitudes can be influenced by noise. So, for a finer tuning, we can add a real constant $\alpha > 0$ in equation (4.25), resulting in:

$$\frac{F_2[v]}{A_2} \approx \alpha \frac{F_1[v]}{A_1} \ \ \Leftrightarrow \ \ \alpha^{-\frac{1}{2}} \frac{F_2[v]}{A_2} \approx \alpha^{\frac{1}{2}} \frac{F_1[v]}{A_1} \tag{4.26}$$

Thus, based in the Euclidean distance, we can define

$$D\{F_1, F_2\} = \sum_v \left(\alpha^{\frac{1}{2}} \frac{F_1[v]}{A_1} - \alpha^{-\frac{1}{2}} \frac{F_2[v]}{A_2} \right)^2 \tag{4.27}$$

as a distance between F_1 and F_2 which results in values near to zero when equation (4.26) is a good approximation (for the cases where F_1 and F_2 are similar) and larger values otherwise.

The regions of low amplitude in functions $F_1[v]$ and $F_2[v]$ suffers more from the noise influence than regions with high amplitude. So, we can add a weight w_v to equation (4.27) in order to

give more importance for the regions of high amplitude, resulting in

$$D\left\{F_1, F_2\right\} = \frac{\sum_v w_v \left(\alpha^{1\!/\!2}\frac{F_1[v]}{A_1} - \alpha^{-1\!/\!2}\frac{F_2[v]}{A_2}\right)^2}{\sum_v w_v} = \frac{1}{W}\left(\alpha\frac{V_{11}}{A_1^2} + \frac{1}{\alpha}\frac{V_{22}}{A_2^2} - 2\frac{V_{12}}{A_1 A_2}\right),\qquad (4.28)$$

$$V_{11} = \sum_v w_v F_1[v]^2,\ \ V_{22} = \sum_v w_v F_2[v]^2,\ \ V_{12} = \sum_v w_v F_1[v]F_2[v]\ \ \text{and}\ \ W = \sum_v w_v,$$

where w_v is defined as:

$$w_v = max\left(\frac{F_1[v]}{A_1}, \frac{F_2[v]}{A_2}\right).\qquad (4.29)$$

The value of $\alpha > 0$ is found as the one minimizing the distance metric $D\left\{F_1, F_2\right\}$ and can be calculated by deriving equation (4.28) as a function of α and equaling it to zero, leading to:

$$\alpha^2 = \frac{A_1^2}{A_2^2}\frac{V_{22}}{V_{11}}.\qquad (4.30)$$

Substituting the value of α given by equation (4.30) in equation (4.28) and simplifying, produces:

$$D\left\{F_1, F_2\right\} = 2\frac{\sqrt{V_{11}V_{22}} - V_{12}}{A_1 A_2 W}.\qquad (4.31)$$

A similarity metric yielding values between 0 and 1, where 1 means completely similar and 0 completely different, can be calculated using a Gaussian function as:

$$S\left\{F_1, F_2\right\} = e^{-\frac{D\{F_1,F_2\}}{2\sigma_s^2}},\qquad (4.32)$$

for a given σ_s.

4.4.2 Algorithm

Now that we have a distance metric capable of comparing the similarity between two Fourier transforms we can use it as a base for a clustering algorithm. This algorithm should be capable of grouping together regions that have similar Fourier transforms, indicating that they carry the same signal.

The algorithm proposed in this work is a modification of the K-means [71]. The K-means algorithm is used to cluster a set of input vectors into k groups (or clusters). Each group is defined by its centroid and the vectors are assigned to the closest cluster. The k centroids are initialized

by a set of representative samples or random selected from the input set and each element of the input set is assigned to the closest centroid. After the elements are assigned, the centroids are updated. This process is repeated iterativally till the centroids converge (or when the difference of the current centroids to the previous ones is below a threshold).

One disadvantage of the K-means for our purpose is that we need to know, prior to the clustering, how many cluster we want to form. The optimal number of cluster is video dependent because the noise from different micro-regions tend to differ from one site to another.

The proposed algorithm is represented in Figure 4.8 and presents two majors modifications:

1. We employ Equation (4.31) as a distance metric;

2. The algorithm decides automatically the optimal number of clusters.

| a) Initial set | b) First iteration | c) After convergence |
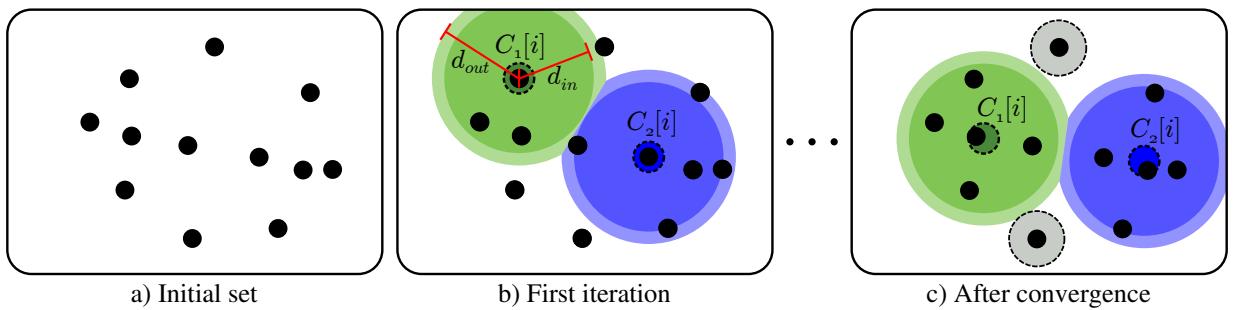
Figure 4.8: Clustering algorithm: a) Initial set of points that we want to cluster; b) Algorithm after first iteration. The points within one of the colored regions belong to the same cluster; c) Algorithm after convergence. The points inside the dashed regions belong to a special cluster where its elements are too different from others clusters.

The input of the algorithm is $\Upsilon = \{F_n[i]\}$, a set of vectors of the Fourier transform for each micro-region, in a total of N_R vectors. This set is also seem as points in a nonlinear space which have Equation (4.31) designed as a distance metric.

Differently from the K-means algorithm, the k-th cluster is entirely defined by the neighborhood around the vector $C_k[i]$, the cluster centroid, that is computed by the mean of the vectors inside the cluster after each iteration. That is, we do not need to know the centroid of other clusters to decided if a vector belongs or not to a given cluster.

In order for a vector to be included in the cluster, its distance to the cluster center must be less or equal to d_{in}. Once inside the cluster, the vector will only be excluded when, in the next update of the cluster center, its distance to the center is superior to d_{out}.

From the initial set Υ, 20% of vectors are randomly selected as cluster center. For all the remaining points we calculate its distance to $C_k[i]$. If this distance is below or equal to d_{in}, the point is integrated to the cluster. Otherwise, if there are no cluster for which the distance is less or equal to d_{in}, the element remain unclassified.

After this step we calculate the mean value of all vectors within a cluster and this mean is used to update the cluster center. As it is possible that in the random selection of elements as cluster

centers that we have separated elements that are in fact very similar, we compute the distance between between all cluster centers to determine how similar they are. If this distance is less or equal to d_{out} the clusters are aggregated together. After this step, for those clusters that have just one element, its vector are aggregated together in a special cluster for those elements that are too different from others. Normally this correspond to elements composed primarily by noise.

It is possible that after updating the cluster center, some elements within a cluster will actually no longer belong to the cluster. Therefore we calculate the distance of all elements within a cluster to the cluster center. Those that have a distance higher than d_{out} are excluded and set as unclassified.

If we still have unclassified elements, we randomly select 20% of the unclassified elements to form new clusters and we repeat the previous steps till there are no more unclassified elements remaining. At this point, the algorithm is said to have achieved convergence.

For some cases it is possible that the algorithm never converges or converge just after a large number of iterations. Hence, we fixed a limit of 200 iterations and the algorithms stops, even without convergence, when reaches this limit.

In this chapter we analysis the performance of our algorithms to real and synthetic data. Given the lack of video databases for the purpose of HR estimation, we created our own database to test the algorithms' performance.

Section 5.1 present the database. The remaining sections evaluate the algorithms' performance.

5.1 DATABASE

The database is composed of video sequences with a duration of 60 seconds. The videos are stored without compression since the motion compensation employed in compression algorithms can eliminate the subtle variations of the skin tone that we employ for HR estimation.

We captured 2 different video sequences from 20 volunteers with an approximated duration of 65 seconds. The videos were cropped to present an exact duration of 60 seconds. Among the volunteers, 15 were man and 5 were women, 4 used glasses and 6 had beards. Figure 5.1 sketch the average face of all volunteers as captured on the first frame of the videos.

a) Steady   b) Movement
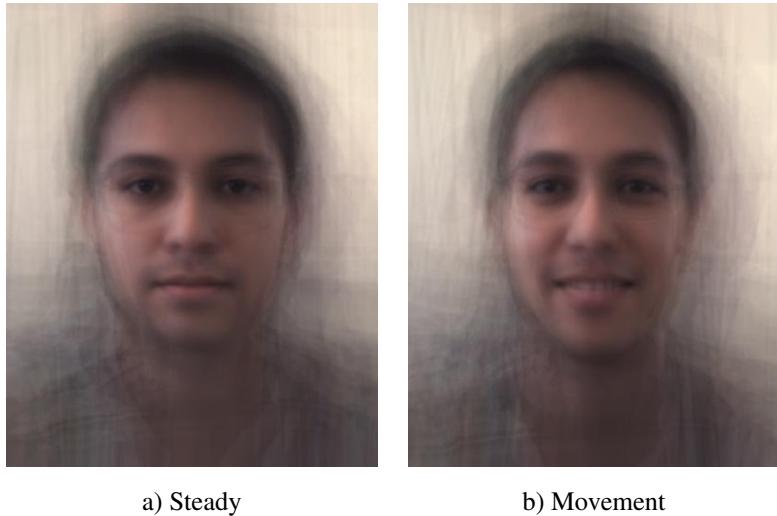
Figure 5.1: Average of volunteers face from the first frame of the videos. The images were aligned with respect to the eyes and mouth position.

The videos were captured using the camera Firefly MV FMVU-03MTC from Point Grey, with a resolution of 0.3 megapixels (640x480 pixels) and a temporal sampling of 60 frames per second. Each frame is captured and stored without compression by a personal computer running an application that communicates to the camera through USB. The frames are stored as a raw data.

During the acquisition process, when the application was not able to gather a frame from the camera buffer before the camera start to overwrite it, some frames were lost. As this error occurs with a relative small frequency, ours algorithms were able to estimate the HR even under such errors. The performance of ours algorithms, evaluated with this database, is shown in subsequent sections.

The captured frames present a Bayer pattern [72] of type 'RGGB' with 8 bits per pixel. The frames are demosaiced using the function `demosaic`[1] from MATLAB.

The camera captured mainly the volunteer's face under two conditions: for the first video, we asked the volunteer to stay as still as possible; for the second video, the volunteer was let to move freely and we interviewed them to encourage movements of the lips, face and hands. The audio was not recorded. The volunteers were not instructed on how to move on the second video in order to obtain a more natural reaction and do not introduce a bias.

As the videos were captured, we monitored the volunteers HR using a commercial fingertip pulse oximeter that presents an accuracy of 2 BPM and a resolution of 1 BPM, as informed by the manufacturer. This measure is used as a reference (the ground-truth) to determine how well the algorithms perform. The oximeter data is filtered with a Gaussian filter with variance equal to $0,7071$ to eliminate its descontinuities due to the low precision, as shown in Figure 5.2.
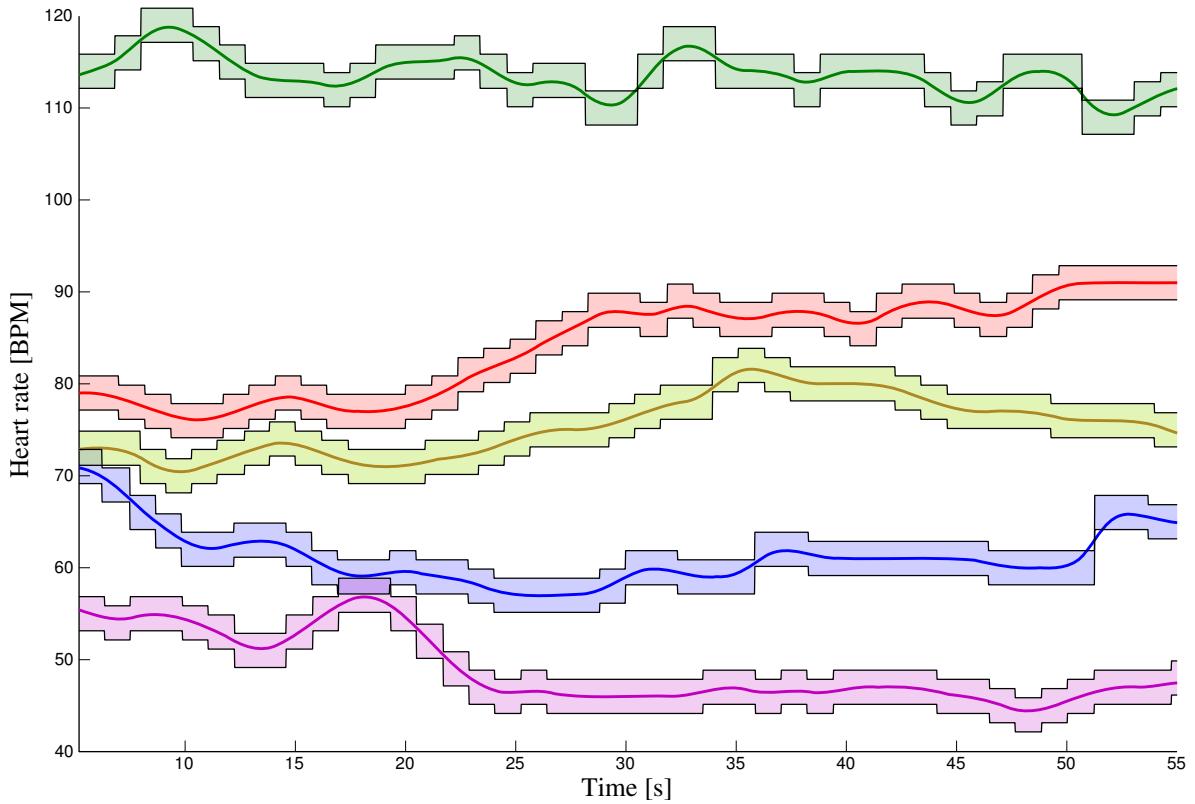
Figure 5.2: Oximeter data filtering. The colored regions correspond to the fingertip oximeter reading for 5 volunteers, within a range of ± 1.86 BPM. The lines show the filtered data for each volunteer.

---

[1] http://www.mathworks.com/help/images/ref/demosaic.html

5.2 EVALUATION OF HR-FD

To evaluate the performance of the algorithms, we employ, in this work, bar plots similar to the one in Figure 5.3. These bar plots show the amount of time, as a percentage from the total time, that the algorithm stayed within a given absolute error range for all 20 volunteers. The error is considered to be the difference between the filtered oximeter reading and the estimated HR.

If the absolute error of the estimated HR, compared to the oximeter reading, is inferior or equal to 2 BPM, then the estimation is considered correct as it falls in the accuracy range of the oximeter. We also assume that errors inferior to 8 BPM are considered acceptable for HR estimations. On the other hand, if the error is superior to 11 BPM, we consider that the estimated HR is incorrect. Hence, the green regions correspond to correct estimations, red regions to incorrect estimations and yellow regions correspond to the transition between acceptable and incorrect.

In this section we evaluate the performance of HR-FD compared to the algorithm of Poh [13]. To perform the analysis we employ the steady videos and the videos with movement from our database, as well as synthetic data.

5.2.1 Steady Videos

Figure 5.3 shows the performance of the proposed algorithm compared to that of Poh. It can be noticed that the HR-FD performs better than POH for the steady videos where Poh stayed 73.9% of the time with an error less or equal to 8 BPM, compared to 86.6% for HR-FD.
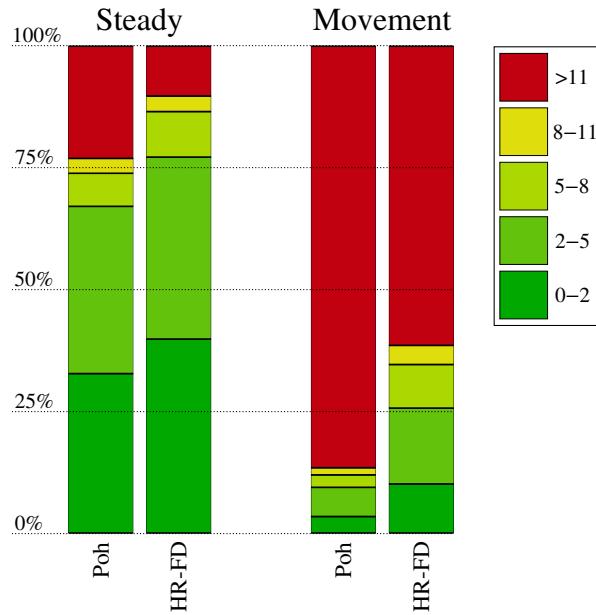

Figure 5.3: HR detection performance of Poh versus HR-FD.

This better performance is due to three factors: the use of the adaptive filter, the derivative filter and the fixed mixture of the traces to form the signal employed for HR estimation. As we apply a fixed mixture for source separation, we are not susceptible to errors that could be introduced

by the Independent Component Analysis algorithm that adaptively estimate the alpha values to apply to each color channel. Also, the adaptive and derivative filter are capable of increasing the SNR before heart rate estimation.

Table 5.1 portray the contribution of the adaptive and derivative filters, and the Blind Source Separation (BSS) strategy: fixed mixture of the red, green and blue channels (the strategy employed in HR-FD); adaptive estimation of the weights to apply to each color channel by means of ICA (the strategy employed by Poh).

Table 5.1: Evaluation of the contribution of the adaptive filter, derivative filter and BSS strategy on HR-FD for the steady videos.

| Filter | | | Error range | | | | | |
Adaptive	Derivative	BSS	$\vert\epsilon\vert \leq 2$	$\vert\epsilon\vert \leq 5$	$\vert\epsilon\vert \leq 8$	$\vert\epsilon\vert > 11$	Position	
No	No	Fixed	40.1%	79.4%	89.3%	07.4%	1st	
		Adaptive	31.8%	65.0%	72.4%	25.7%	7th	← Poh
	Yes	Fixed	38.1%	72.9%	81.1%	15.5%	5th	
		Adaptive	28.6%	58.1%	64.4%	34.1%	8th	
Yes	No	Fixed	40.8%	78.2%	87.3%	09.5%	2nd	
		Adaptive	31.8%	66.1%	73.9%	22.8%	6th	
	Yes	Fixed	39.8%	77.2%	86.6%	10.2%	3rd	← HR-FD
		Adaptive	36.6%	73.0%	81.8%	14.0%	4th	

The best performance was achieved when we used the fixed mixture for BSS and when no adaptive and derivative filters were employed. The HR-FD obtained the third place, but we can observe that its performance is very close to first and second place. The algorithm of Poh, on the other hand, obtained the seventh place.

It can be seen from the average contribution, depicted in Table 5.2, that the use of the adaptive filter and the fixed mixture improved the performance of the algorithm. The use of the filters did not provide the best performance for the steady videos since their function is to attenuate noise and the signal captured on this case comprises a good SNR as little motion artifact is present. However, they provided a performance close to the best, achieving the second and third position with only 2.0% and 2.7% less correct estimated HR than the first place, respectively.

Table 5.2: Average contributions of the adaptive filter, derivative filter and BSS strategy on HR-FD for steady videos. The average percentage of incorrect estimated HR ($\vert\epsilon\vert > 11$) and the average position is shown in each cell.

| | Employed | |
Feature	Yes	No
Adaptive Filter	14.1% (3.75th)	20.7% (5.25th)
Derivative Filter	18.5% (5.00th)	16.3% (4.00th)
Fixed Mixture BSS	10.7% (2.75th)	24.1% (6.25th)

5.2.2 Videos with Movement

For the videos with movement, the difference between the performance of HR-FD and Poh is even higher than for the steady videos, as it can be seen in Figure 5.3, resulting in 11.9% correct estimations for Poh and 34.6% for HR-FD.

For these videos, the use of the adaptive and derivative filters become more important because they present a lower SNR, resulting in an improvement of the algorithm performance, as show in Table 5.3. Indeed, the best performance is acquired when both filters are employed, with an advantage of 6.4% and 7.9% more correct HR estimations than the second and third place. The average contribution, shown in Table 5.4, enforces this statement. As the use of the adaptive and derivative filters provide a large advantage for the videos with movement and a performance close to the best for the steady videos, they are employed in our algorithm for HR estimation.

Table 5.3: Evaluation of the contribution of the adaptive filter, derivative filter and BSS strategy on HR-FD for the videos with movement.

Filter			Error range					
Adaptive	Derivative	BSS	$|\epsilon| \leq 2$	$|\epsilon| \leq 5$	$|\epsilon| \leq 8$	$|\epsilon| > 11$	Position	
No	No	Fixed	07.9%	20.3%	26.1%	71.7%	4th	
		Adaptive	02.6%	07.5%	09.3%	89.7%	8th	$\leftarrow$ Poh
	Yes	Fixed	06.8%	16.9%	23.0%	74.5%	5th	
		Adaptive	04.0%	13.9%	18.4%	78.7%	6th	
Yes	No	Fixed	07.9%	20.1%	28.1%	68.6%	2nd	
		Adaptive	03.4%	10.4%	14.4%	83.9%	7th	
	Yes	Fixed	10.1%	25.6%	34.5%	61.6%	1st	$\leftarrow$ HR-FD
		Adaptive	06.7%	20.7%	26.6%	71.2%	3rd	

Table 5.4: Average contributions of the adaptive filter, derivative filter and BSS strategy on HR-FD for the videos with movement. The average percentage of incorrect estimated HR ($|\epsilon| > 11$) and the average position is shown in each cell.

	Employed	
Feature	Yes	No
Adaptive Filter	71.3% (3.25^{th})	78.6% (5.75^{th})
Derivative Filter	71.5% (3.75^{th})	78.5% (5.25^{th})
Fixed Mixture BSS	69.1% (3.00^{th})	80.9% (6.00^{th})

As it can be seen, the use of the adaptive BSS is the main factor that reduces the performance of the algorithm. The application of the fixed mixture, combined with the adaptive and derivative filters, provided a better robustness against noise.

5.2.3 Synthetic Data

Since it is hard to evaluate the noise performance of the algorithms for a real scenario, as signal and noise energy are unknown, we created a simulated experiment to investigate their performance against noise.

The PPG signal was simulated as a cossinus wave of known frequency, randomly chosen between 60 and 200 BPM (uniform random variable) added to a Gaussian noise of known energy. Figure 5.4 depicts the performance of HR-FD and Poh's algorithm to this simulated data. We also tested the performance of HR-FD without the adaptive and derivative filters to evaluate their contribution.
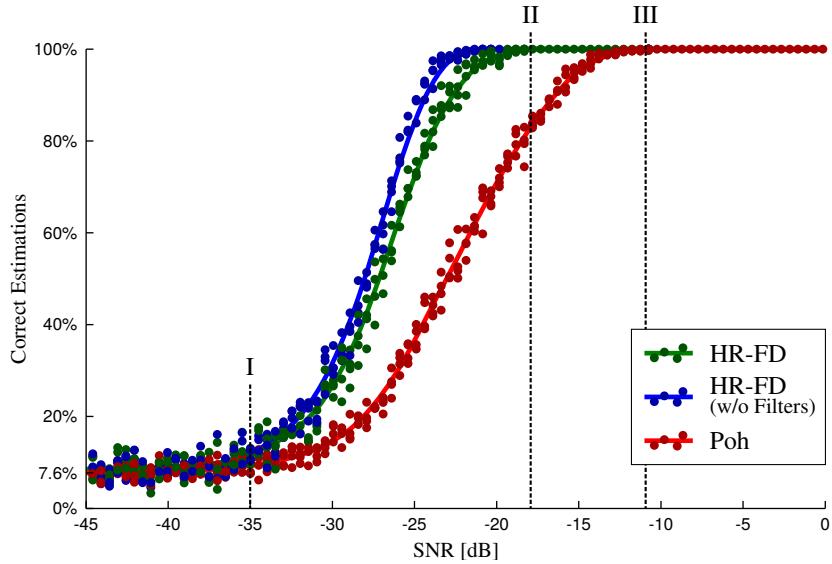

Figure 5.4: Evaluation of algorithm performance to Gaussian noise. A synthetic signal composed by a cossinus wave of kwon frequency plus a Gaussian noise is fed to the algorithms. The percentage of time the estimation error is less or equal to 8 BPM is plotted for different SNR values. Each point on the curve is the average value obtained in 100 simulations.

When the SNR is below $-35\ dB$ (point I), both algorithms present an equal percentage of correct estimations equivalent to that of randomly choosing the output. Indeed, it can be shown (see Appendix II) that the probability of percentage of time that the absolute error $|\epsilon|$ is inferior or equal to T_{BPM} (for $T_{BPM} \leq 30$, the error threshold), when the signal frequency and the algorithm estimation are uniform independent random variables, is given by

$$P\left\{|\epsilon| \leq T_{BPM}\right\} = \frac{2T_{BPM}(200 - 60)}{(240 - 30)(200 - 60)} = \frac{T_{BPM}}{105}. \qquad (5.1)$$

For $T_{BPM} = 8$, $P\left\{|\epsilon| \leq 8\right\} = 7.6\%$, that correspond to the values found in Figure 5.4 for $SNR < -35dB$.

Between points I and III we can notice that HR-FD performs significantly better than Poh for this synthetic signal. At point II, where $SNR = -18\ dB$, HR-FD reaches 100% of correctly estimated HR. Poh reaches 100% at point III, where $SNR = -11\ dB$. This means that the

HR-FD achieves the maximum rate $7\ dB$ before Poh, where the noise energy is 5 times greater than the noise energy at point III. Poh reaches 50% for $P\left\{|\epsilon| \leq 8\right\}$ at $-23\ dB$ while HR-FD at $-26.9\ dB$.

This better performance is due to the use of the fixed weights employed for BSS. The synthetic data employed here shows that the use of the adaptive and derivative filters do not contribute to improve the performance. Actually, the performance was slightly better when they were not employed. This is the opposite behavior then the one found for the real data, where the use of the adaptive and derivative filters enhanced the performance. It indicates that the Gaussian noise do not simulate very well the noise found in real data.

From our database we can observe that the noise present higher amplitudes for low frequencies, and the noise amplitudes tend to decay inversely to the frequency, as shown in Figure 5.5. The low frequency characteristic of the noise was also observed by Xu *et. al.* [9].

a) Steady videos   b) Videos with movement
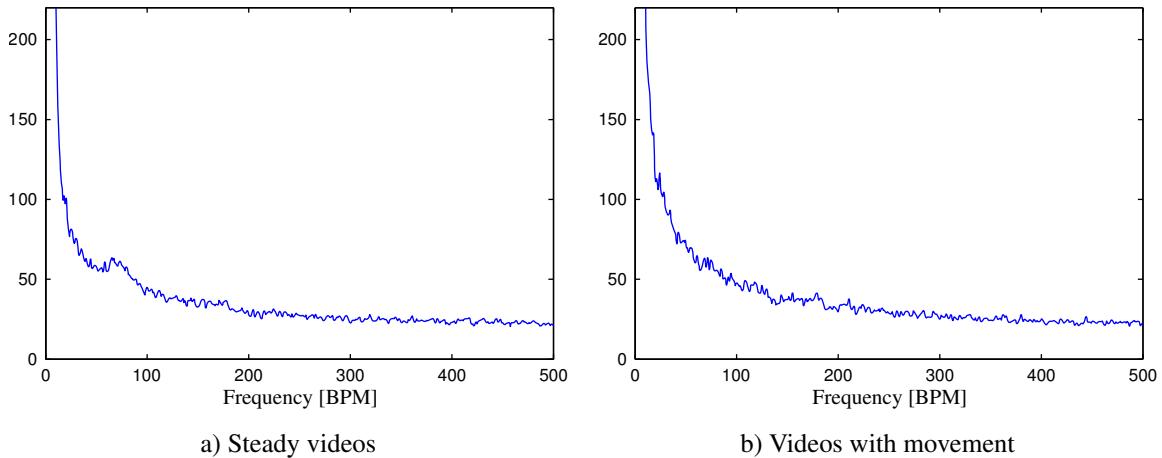
Figure 5.5: Discrete Fourier transform of the noise on real data. The curves depicts the average value of the DFT obtained for all volunteers. The frequencies near the value obtained by the pulse oximeter ($\pm$ 10 BPM), for each volunteer, were not used to compute the average in order to eliminate the PPG signal from the resulting curves.

Therefore, to obtain a simulated noise more befitting to the real data, we integrate the Gaussian noise in time to amplify the low frequencies and attenuate the high frequencies, applying a gain inversely proportional to the frequency. We also remove the DC component of the noise and we multiply it by a constant in order to obtain the desired SNR. The frequency characteristic of the Gaussian noise and the integrated Gaussian noise are shown in Figure 5.6.

With this noise we obtain a different scenario. Results are shown in Figure 5.7 that depicts the percentage of time that the algorithm presented an absolute error inferior to 8 BPM for 100 simulations.

Table 5.5 present the SNR values where which algorithm reached the given percentage of correct estimation. We can notice that HR-FD reach 10% of correct estimations at -39.09 dB, 13.33 dB before the algorithm of Poh, and 95% at -30.18 dB while Poh reach this percentage at -1.79 dB. This better performance is mainly due to the fixed mixture employed for BSS that avoids the errors introduced by ICA at low SNR because this algorithm needs a relative high

a) Gaussian noise

b) Integrated Gaussian noise
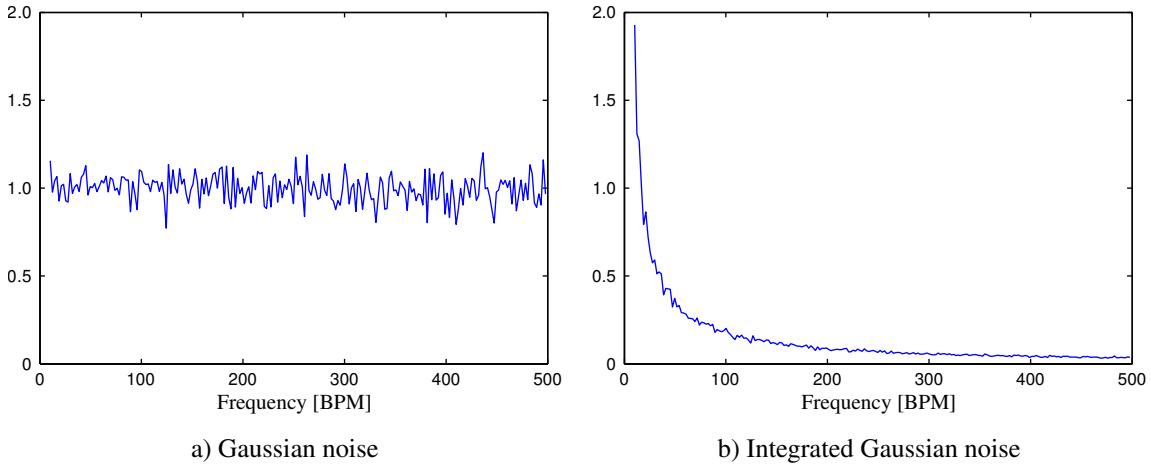
Figure 5.6: Discrete Fourier transform of the simulated noise.
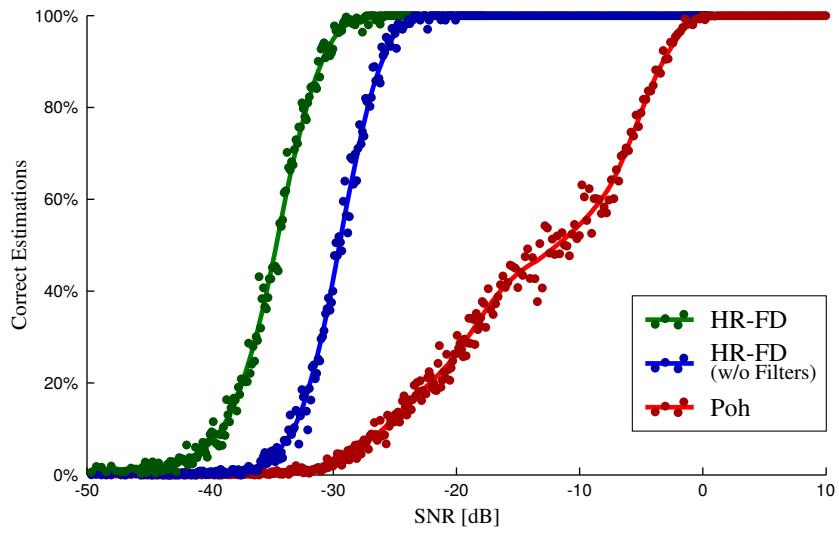
Figure 5.7: Evaluation of algorithm performance to integrated Gaussian noise. A synthetic signal composed by a cossinus wave of kwon frequency plus a Gaussian noise is fed to the algorithms. The percentage of time the estimation error is less or equal to 8 BPM is plotted for different SNR values. Each point on the curve is the average value obtained in 100 simulations.

value of SNR in order to correct estimate the weights to apply for each color channel. This can be observed by the performance of the algorithm employing only the fixed mixture (HR-FD w/o Filters) that presented a better performance than Poh.

Table 5.5: SNR values, in dB, where which algorithm reached the given percentage of correct estimations.

	10%	50%	95%
HR-FD	-39.09	-34.75	-30.18
HR-FD (w/o Filters)	-33.25	-29.36	-25.07
Poh	-25.76	-11.89	-01.79

Also, we can observe that the use of the adaptive and derivative filter contributed for a better

performance and the HR-FD reached a given percentage of correct estimations 5.4 dB, in average, before its versions that do not employ those filters.

5.2.4 Conclusion

The HR-FD presented an improvement in performance, compared to the algorithm of Poh, both in the videos with and without movement and on the synthetic data. We noticed that this better performance is mainly due to the use of the fixed mixture employed for BSS. The adaptive and derivative filters also contributed and an enhancement in performance was observed with their use.

5.3 EVALUATION OF HR-MRT

Nonetheless, HR-FD and Poh do not show a satisfactory performance for videos with movement, as was already stated in Section 3.2, due to movement artifacts. This issue is addressed by HR-MRT, that employs a more complex trace extraction.

The HR-MRT is tuned by parameters that influence, for example, the frame segmentation in micro-regions, point tracking filtering, clustering algorithm, etc. Table 5.6 summarizes the parameters employed for HR-MRT. Sections 5.3.1, 5.3.2 and 5.3.3 disclose the choice of the main parameters. In Section 5.3.4 we compare the performance of HR-MRT to HR-FD and Poh.

Table 5.6: Parameters employed for HR-MRT.

Stage		Parameters
Frame Segmentation	Bilateral Filter	$\sigma_1 = 3$ (control the distance penalty) $\sigma_2 = 0.06$ (control the color difference penalty) Squared neighborhood of size 11x11
	Low-pass Filter	$N = 6$ (kernel size)
Skin Detection		Histogram based approach combined with Viola-Jones face detector
Block Duration		10 seconds
Point Tracking Filtering	Temporal Filter	Third order polynomials
	Spacial Filter	Full affine transformation
Clustering Algorithm		$d_{in} = -2\,ln(0.4)$ (similarity of 0.4 for $\sigma_s = 1$) $d_{out} = -2\,ln(0.42)$ (similarity of 0.42 for $\sigma_s = 1$)
Traces Prefilter		Employs the adaptive and derivative filters

5.3.1 Block Duration

One block correspond to a set of frames upon each we perform point tracking in order to compensate for movements. The smaller the block duration the less the algorithm will be affected by drift errors and occlusions on the video when, for example, a moving hand occlude part of the volunteer's face. On the other hand, blocks can introduce artifacts due to discontinuities in the transition from one block to another. Therefore, too small blocks should also be avoided. In our work, the traces of each micro-region are normalized, removing its DC component and adjusting its amplitude, resulting in a signal with unitary variance. This normalization reduces the discontinuities between blocks, but it does not eliminate them.

In Figure 5.8 we show the performance of the HR-MRT for two block durations: 60 and 10 seconds. The performance was better for the estimations using blocks of 10 seconds, compared to the one of 60 seconds. This better performance is due to the point tracking that can get lost when applied for long sequences.
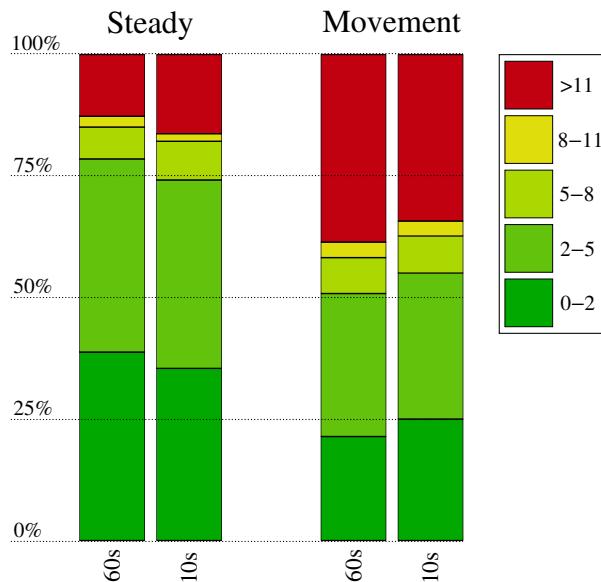
Figure 5.8: Performance of HR-MRT for different block duration.

The opposite behavior is found for the steady videos, where blocks of 60 seconds perform better than blocks of 10 seconds. As there is little movement, execute the point tracking using blocks of 60 seconds is not very challenging and the algorithm offers, in general, a very good estimation while avoiding the artifacts introduced in the transitions from one block to another.

As the videos with movement present a more challenging scenario, we decided to employ blocks of 10 seconds. This block duration showed the best performance for videos with movement and a satisfactory performance for steady videos.

5.3.2 Skin Detection

The skin detection plays an important role in the definition of the ROI, eliminating the pixels from the background. Although, skin detection is a hard task as the color of skin pixels overlap the color of non-skin pixels found in the nature. Therefore, we proposed three strategies, as described in Section 4.1:

- **Auto:** Automatic skin detection using a histogram based approach;

- **Viola-Jones:** The result of the automatic skin detection is combined with the Viola-Jones face detector to eliminate those pixels that are not on the volunteers face;

- **Manual:** The volunteer skin was manually selected on the first frame of the video.

Figure 5.9 shows the performance of the three strategies for two block duration.
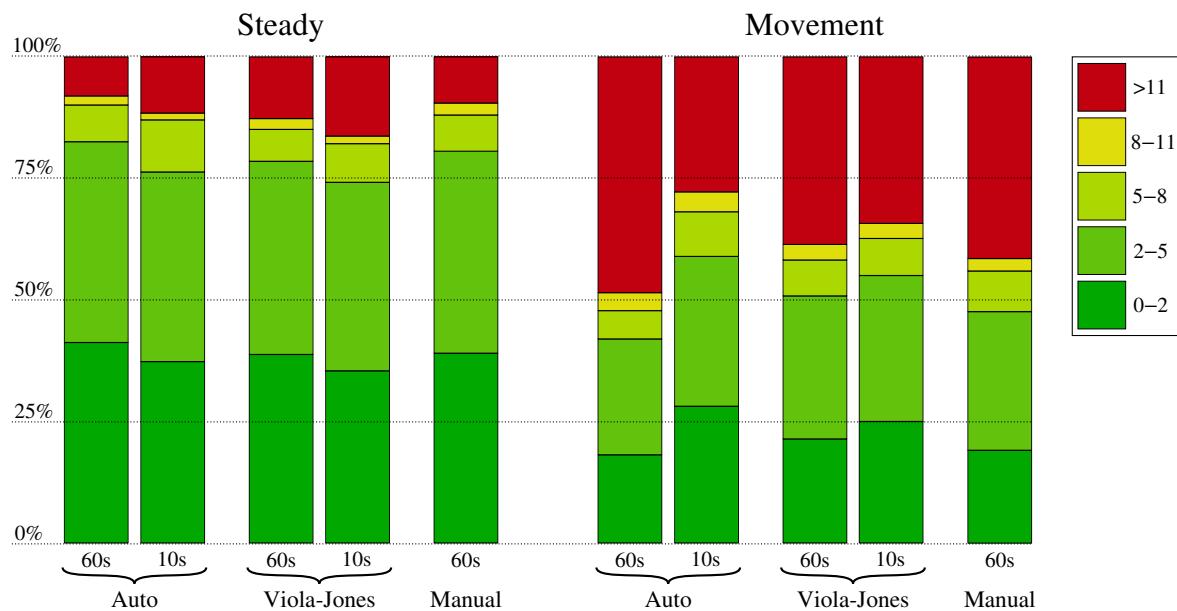
Figure 5.9: Performance of HR-MRT for different skin detection strategies.

The use of automatic skin detection presented the best performance when combined with blocks of 10 seconds, while for blocks of 60 seconds the automatic skin detection combined with the Viola-Jones face detector had the best performance. Manual detection were only tested for blocks of 60 seconds and did not offer the best performance because, as it can be seen in Figure 4.2, it is just a rough contour that includes eyes, eyebrows, teeth and parts of clothing that negatively influence the estimation.

5.3.3 Point Tracking Filtering

Points are tracked using eight successive frames of the video, as schematized in Figure 4.6. The output of the tracking algorithm is considered rather an estimation of the the actual optical

flow and some noise may be added. To reduce the noise we exploit the temporal and spatial redundancy. The temporal redundancy is used by modeling the movement of the tracking points by polynomials. The position of the estimated tracking position is adjusted in order to fit to the model. The spatial redundancy is explored by means of an affine transformation, that combine together up to 12 points within a small region (the micro-region).

The affine transformation is executed in two distinct ways:

- **Rigid:** Using only rigid transformation: translation, scaling and rotation;

- **Full:** Using a full search: translation, scaling, rotation, shearing and mirroring.

The performance of the different affine transform settings is presented in Figure 5.10. Their performance is, in fact, very similar and it is hard to decide which is better. These results indicate that the rigid transformation can, actually, represent well the movements of the micro-regions. The full search would, in this case, result in parameters close to that for rigid transformation, what implies in similar performance
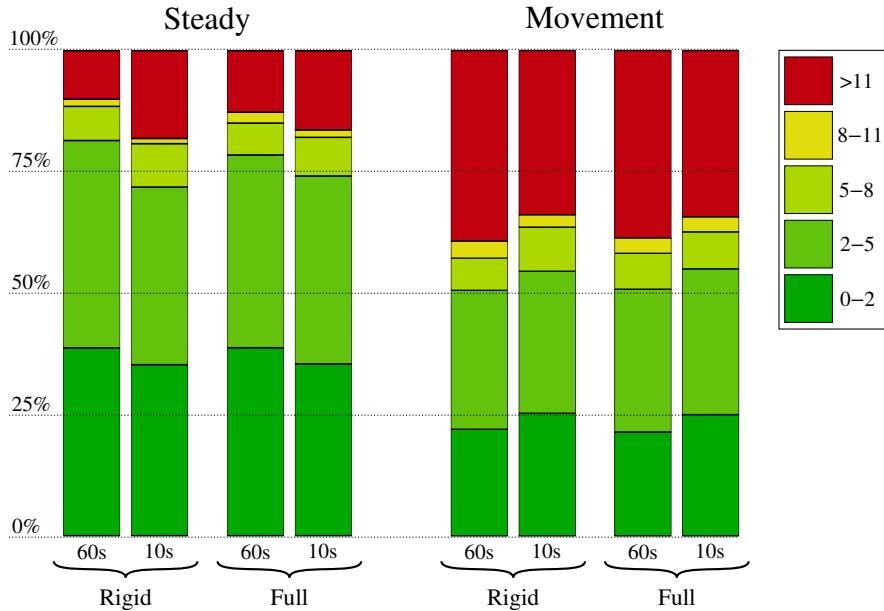
Figure 5.10: Performance of HR-MRT for different affine transform settings.

The polynomial used to model the movement vary in order. From the scheme used, it can be used polynomial of zeroth to sixth order. Figure 5.11 depicts the performance for polynomial from second to fourth order.

Third order polynomials is the one that presented the best performance among the three of them for videos with movement. The use polynomials of smaller order is not very well suited because they can not model satisfactorily the movement of the tracking points. Higher order are also not a good choice because, despite their ability to model complex movements, they do not offer a good noise suppression as the noise can fit more easily to it.
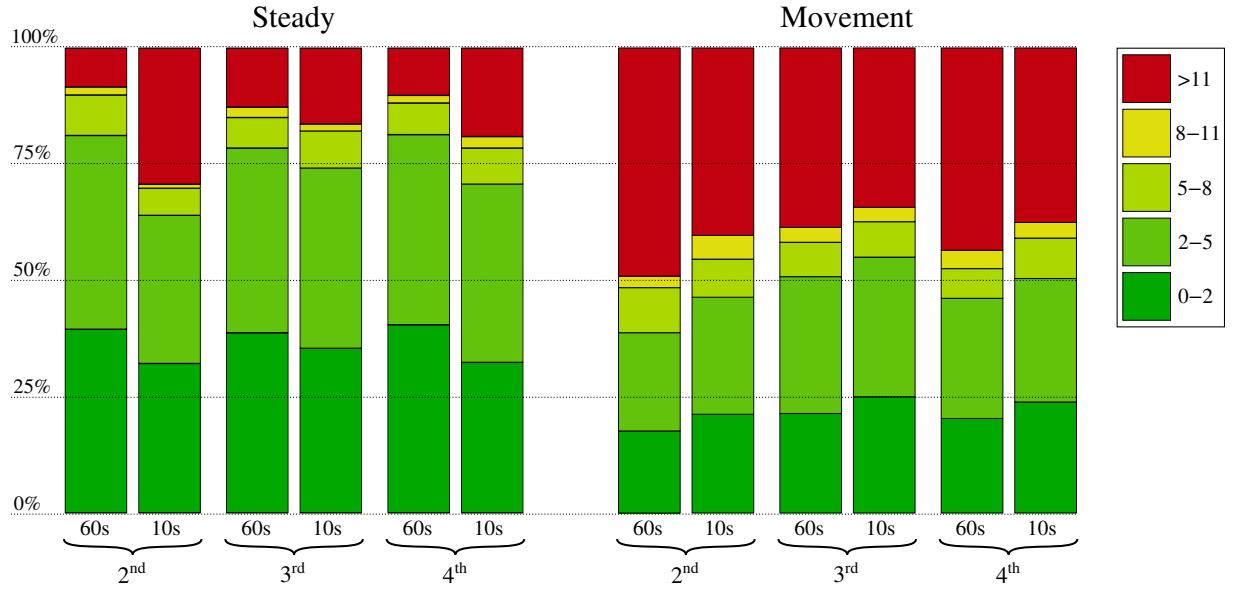
Figure 5.11: Performance of HR-MRT for different polynomial orders.

5.3.4 HR-MRT Performance

Figure 5.12 presents a comparison between the algorithm of Poh, HR-FD and HR-MRT. The HR-MRT presents a significant gain in performance compared to Poh and HR-FD for the videos with movement, where the point tracking is used to reduce artifacts introduced by movement.
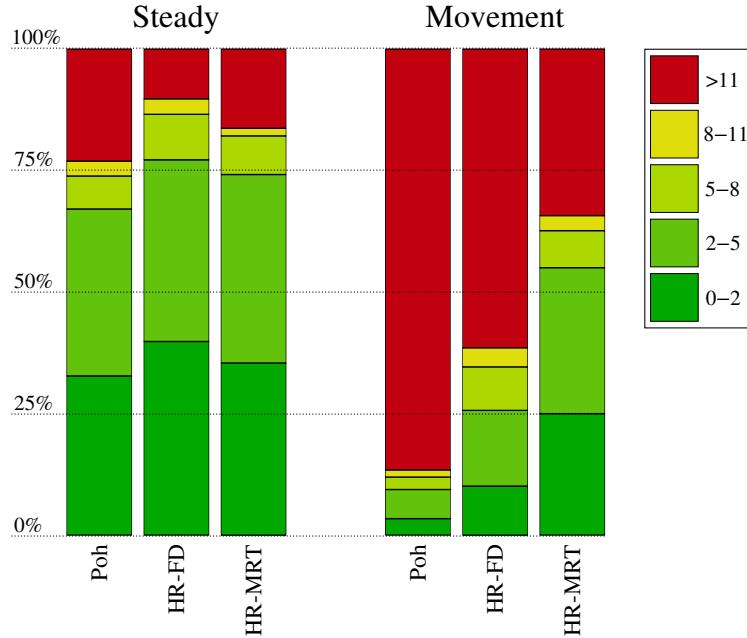
Figure 5.12: Performance of HR-MRT compared to HR-FD and Poh.

Although, for the steady videos, the performance of the HR-MRT is slightly inferior to that of HR-FD but still superior to Poh because of how the ROI is defined. For the HR-MRT we employ a skin detector and point tracking. Therefore, it is possible that some micro-regions on the volunteers face, that could be used for HR detection, were rejected either by the skin detector

or the point tracker before the computation of the red, green and blue traces. Indeed, for the videos on our database, some frames were lost during data acquisition. The discontinuities introduced by these errors are not very well modeled by third order polynomials, which can introduces a drifting effect that may culminate in the rejection of the micro-region.

This effect also occurs for the videos with movement, but most of the rejected regions on those videos are affected by noise. Therefore, eliminating them actually results in a better SNR.

In general, we can conclude that the HR-FD is the best algorithm among the three shown in Figure 5.8 to be employed for videos with little movement. When movement is present the HR-MRT provides better estimations and the use of small blocks is preferable.

5.4   ANALYSIS OF THE AUTOMATIC ROI SELECTION

The HR-MRT divides the frames of the video in micro-regions. The micro-regions that do not meet the criteria imposed during skin detection, point tracking and clustering are eliminated. These criteria are:

1. **Skin detection:** those micro-regions that do not contain at least 80% of pixels detected as skin are reject as the PPG signal can only be extracted from skin pixels;

2. **Point tracking:** the micro-regions for which the affine transform result in a area superior to twice or inferior to half of the initial area in a given time are rejected as their estimated optical flow may contain errors;

3. **Clustering:** we keep only the cluster of micro-regions, grouped accordingly to their DFT, that presents the highest number of elements. The micro-regions of the other clusters are eliminated as they are likely to be affected by motion noise.

The remaining micro-regions are used for HR estimation and form the ROI. Figure 5.13 sketch the percentage of time that regions on the video were chosen to compose the ROI.

The forehead and cheeks were the most chosen as they correspond to very well vascularized regions that facilitate the extraction of the PPG signal in the case of steady videos. For the videos with movement, the algorithm gives privilege to the forehead over the cheeks as the cheeks contain, in general, more motion noise in this case. We can also observe that eyes and mouth were rejected most of the time, mainly for videos with movement, as they introduce artifacts due to speaking and blinking. In the case of automatic skin detection without Viola-Jones face detector we can see that the region on the neck were also employed for HR detection.

The density of chosen regions were higher when using blocks of smaller duration because less micro-regions are rejected during the point tracking stage. This is not crucial for steady videos as a considerable number of micro-regions, in general, are kept after the point tracking. On
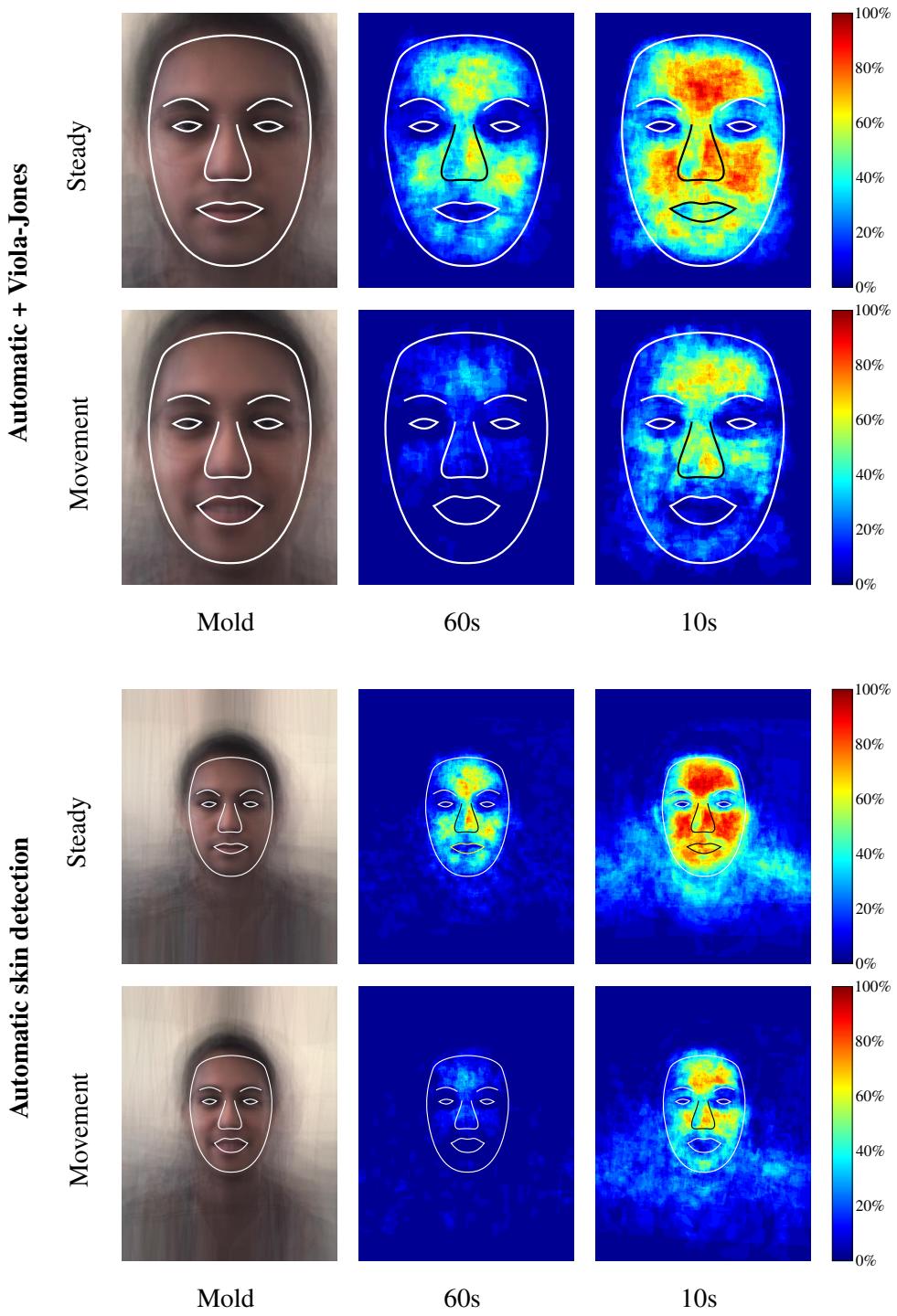
Figure 5.13: Percentage of time that the regions were chosen as ROI by HR-MRT. The mold is based on the average face of the volunteers, after synchronizing them, and depicts the position of face, eyes, mouth and nose. The second and third columns shows the percentage of time that a given region was selected for HR detection for the block duration of 60 and 10 seconds.

the other hand, for videos with movement, the number of rejected regions during point tracking increases, making the use of blocks of smaller duration a better choice. This effect can be seen in Figure 5.14 that depicts the number of micro-region employed for HR detection for each volunteer

individually. We can notice that most of the micro-regions were eliminated during the point tracking for blocks of 60 seconds and, for videos 02 and 05, for example, almost none remained for HR detection.
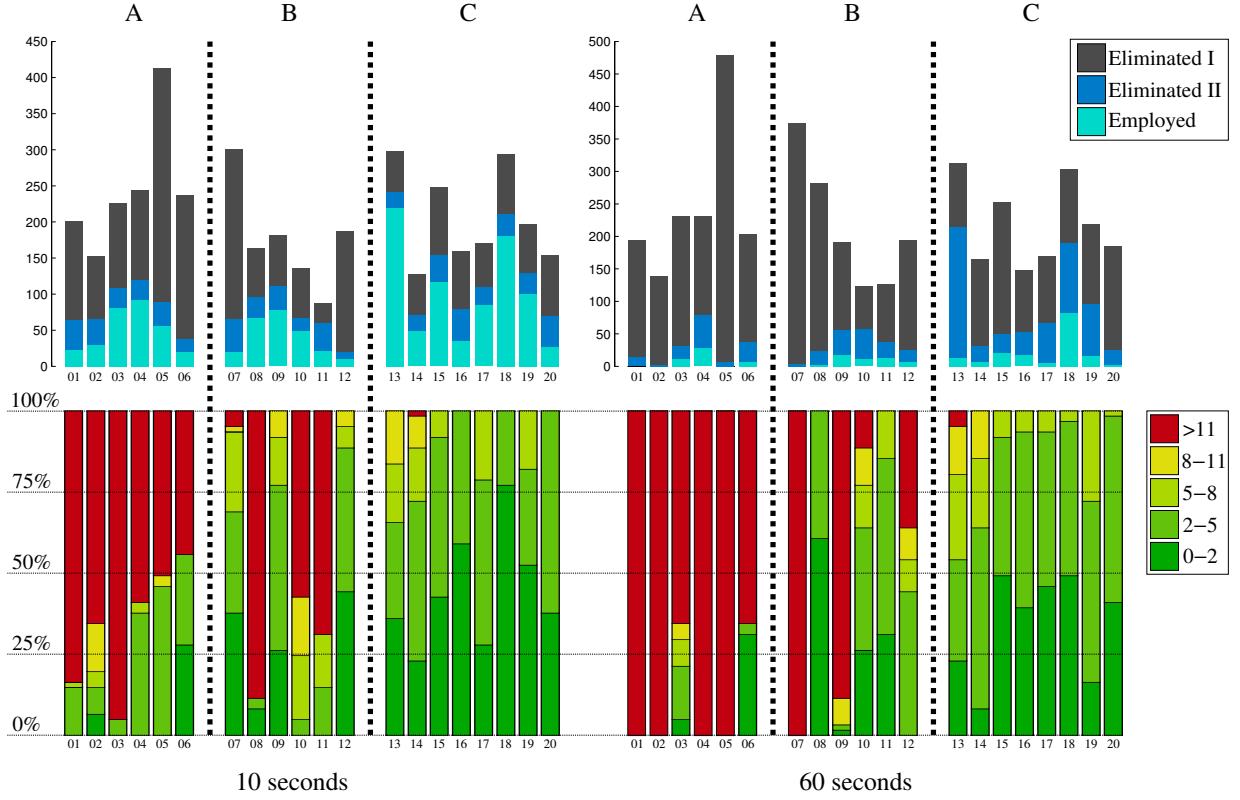
Figure 5.14: Performance of the HR-MRT for the video with movement for each volunteer individually. The first row depicts the number of micro-regions employed for HR estimation in light blue, for two blocks duration. In dark blue is represented the number of micro-regions eliminated by the clustering algorithm and in gray the number of micro-regions eliminated during point tracking. The total height of the bars show the total number of micro-regions initialized by the skin detector. The second row presents the performance separated in three groups: A, B and C.

The volunteers were segregated into three groups, for better visualization, composed by those videos where: A) both the block sizes tested presented a poor performance; B) blocks of 10 and 60 seconds diverged greatly in performance; C) both block sizes presented a good performance. It can be observed that, in general, the higher the number of micro-regions employed for HR detection the better is the HR detection performance. We can also see that for videos with a poor performance, most of the micro-regions were eliminated during the point tracking stage.

5.5 SUMMARY

Table 5.7 summarizes the performance of the proposed algorithms, compared to that of Poh, when using skin detection combined with the Viola-Jones face detector. The HR-FD, despite its simplicity, is the best suited for videos without movement, resulting in only 10.2% of incorrect HR estimations, less than half of the percentage found with Poh's algorithm. For the videos with

movement, on the other hand, HR-MRT presented the best performance with 62.6% of correct estimations, almost twice that of HR-FD and 5.3 times than that of Poh.

Table 5.7: Performance of the algorithms of Poh, HR-FD and HR-MRT.

Video Group	Algorithm	Error range			
		$\lvert\epsilon\rvert \leq 2$	$\lvert\epsilon\rvert \leq 5$	$\lvert\epsilon\rvert \leq 8$	$\lvert\epsilon\rvert > 11$
Steady	Poh	32.7%	67.1%	73.9%	23.1%
	HR-FD	39.8%	77.2%	86.6%	10.2%
	HR-MRT	35.2%	71.9%	80.8%	18.1%
Movement	Poh	03.4%	09.4%	11.9%	86.6%
	HR-FD	10.1%	25.7%	34.6%	61.5%
	HR-MRT	25.0%	55.0%	62.6%	34.3%

Figures 5.15 and 5.16 show the HR estimation over time for two videos without movement. For most steady videos, all three algorithms are capable of correctly estimate the HR, as displayed in Figure 5.15. The proposed algorithms have a better robustness against noise and are capable of maintaining a good performance for more challenging videos, such as the one in Figure 5.16.

Figures 5.17, 5.18 and 5.19 exhibit the HR estimation over time for three volunteers for the videos with movement. Figure 5.17 is an example where none of them were capable to offer a satisfactory HR estimation. Nevertheless we can notice a better performance than the algorithm of Poh due to use of the fixed mixture and the filters employed.

Figure 5.18 depicts an example where all three of them offered a satisfactory estimation. Notwithstanding we can notice that the HR-MRT stayed closer to the ground-truth due to the motion compensation. The effect of the motion compensation is more outstanding in Figure 5.19, where only HR-MRT was capable to correct estimate the HR.
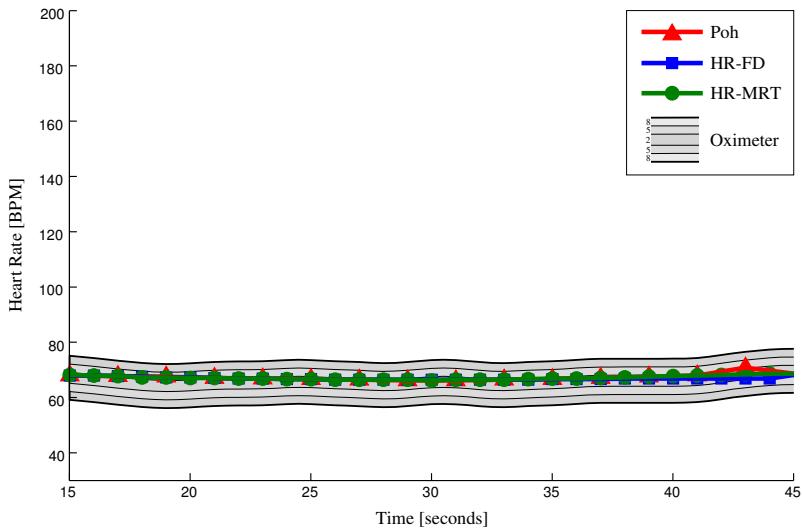
Figure 5.15: HR detection for volunteer 18 - Steady video. The oximeter reading is shown in the range of absolute errors of 2, 5 and 8 BPM.
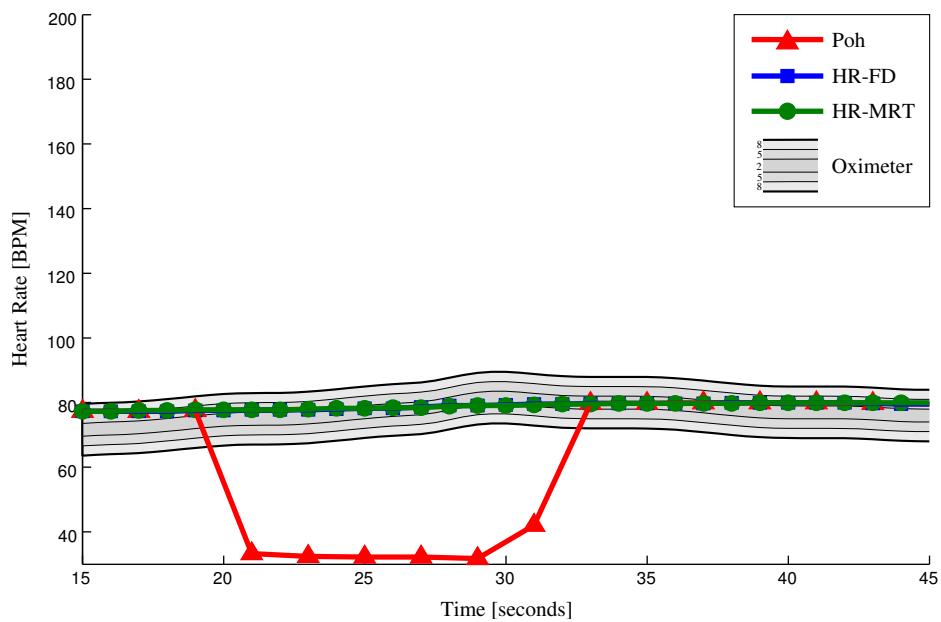
Figure 5.16: HR detection for volunteer 18 - Steady video. The oximeter reading is shown in the range of absolute errors of 2, 5 and 8 BPM.
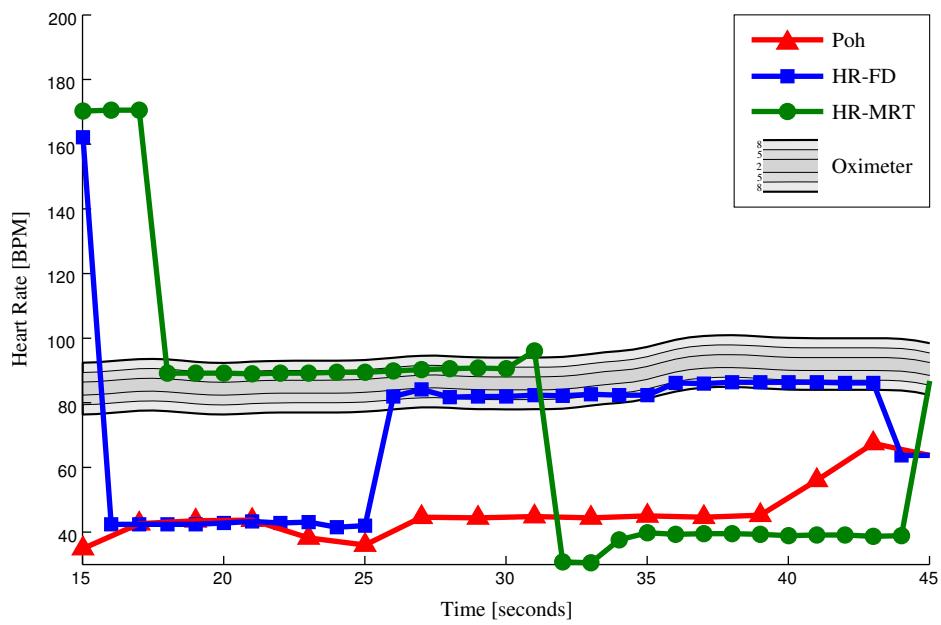
Figure 5.17: HR detection for volunteer 05 - Video with movement. The oximeter reading is shown in the range of absolute errors of 2, 5 and 8 BPM.
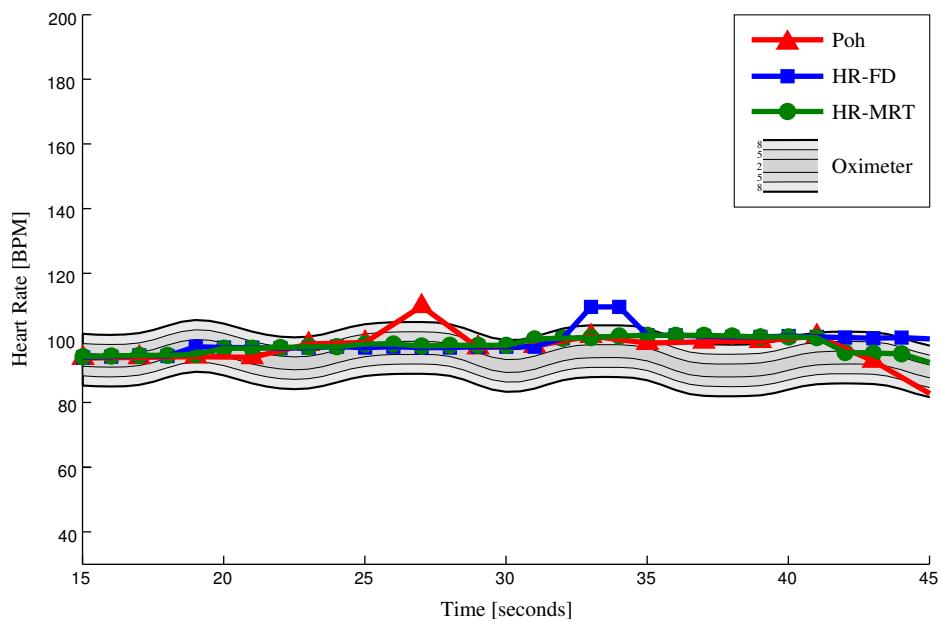
Figure 5.18: HR detection for volunteer 13 - Video with movement. The oximeter reading is shown in the range of absolute errors of 2, 5 and 8 BPM.
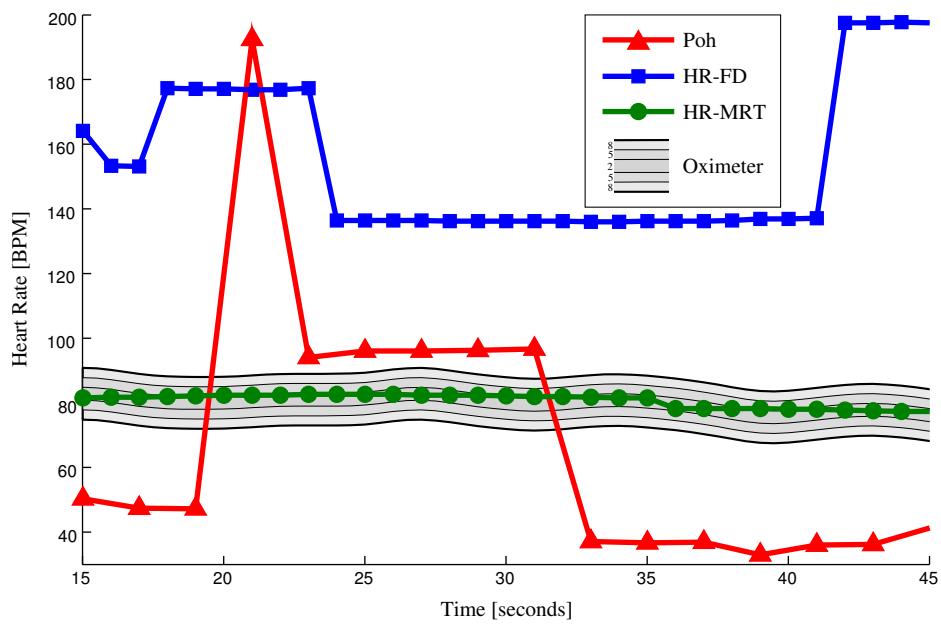

Figure 5.19: HR detection for volunteer 18 - Video with movement. The oximeter reading is shown in the range of absolute errors of 2, 5 and 8 BPM.

6 CONCLUSIONS

In this work we proposed two algorithms for heart rate (HR) estimation using videos of the human face under uncontrolled light in indoor enviroments. We compared our results to that of Poh [13] and observed a substantial improvement.

The first algorithm, Video heart rate estimation through Face Detection (HR-FD), employs an adaptive filter that imposes a temporal coherence on the signal. This filter is based on the assumption that the heart rate varies slowly with time. A derivative filter was also employed to reduce the influence of low frequency noise. These filters boosts the signal to noise ratio of the signal used for HR estimation and we showed that they are capable of reducing the number of incorrect estimated HR, particularly for noisy traces. But predominantly, a larger improvement was obtained by avoiding the use of Independent Component Analysis (ICA), commonly used in the literature. Without ICA we eliminate the dependence on the performance of this algorithm and we are not susceptible to errors that could be introduced when it is not capable to correctly determine the mixture of the three traces. The proposed algorithm presents a low complexity and, despite its simplicity, can correctly estimate the heart rate for videos with little movement.

The second algorithm, Video heart rate estimation through micro-region tracking (HR-MRT), further improved the performance by adding robustness to motion. This algorithm used a different Region of Interest (ROI) than the previous one. The video is divided in blocks and the first frame of the block is segmented in micro-regions using watershed segmentation. For each micro-region we select a set of tracking points that are used to compensate for movement. The optical flow of the tracking points is estimated using the Lucas-Kanade algorithm. The optical flow is spatial and temporal filtered to reduce the effect of noise.

For the temporal filter we modeled the motion of the tracking points with a polynomial function. The order of the polynomial should be high enough to correctly accommodate the complex movements of the tracking points but small enough to avoid data overfitting, which reduces the capacity of the algorithm to eliminate or attenuate noise. The results indicate that the use of third order polynomials presents the best trade-off between data representation and noise attenuation.

The spatial filtering is executed finding the affine transformation that represents the tracking points for each micro-region, minimizing the quadratic error, at a given time. This affine transformation is then applied to determine how the border of the micro-region evolved. We proposed two approaches for the affine transform: restrict the transformation to only rigid transforms and allow a full search. The results suggest that the full search is slightly better than the search restricted to rigid movements in terms of number of correct estimated HR for the case of videos with movement. Although, it is hard to decide which of the two approaches is better and we can conclude that this choice is not crucial for HR estimation.

The HR-MRT algorithm also employs a clustering algorithm to decide which micro-regions

to exploit for HR estimation, automatically defining the ROI. This algorithm is based on K-means and try to cluster those micro-regions that present a similar DFT. For the sake of automation we use only the cluster that presents the largest number of elements based on the assumption that most of the micro-regions will contain the photoplestimographic (PPG) signal and that the micro-regions composed primarily by noise do not cluster very well as the noise vary greatly from one site to another. This assumption is dependent on the performance of the skin detection algorithm, as micro-regions that are incorrectly detected as skin do not contain the desired signal. We observed from the results that to perform the skin detection using the histogram based approach, combined or not with the Viola-Jones face detector, presents a good performance.

6.1 FUTURE WORK

We noticed that, in general, the higher the number of micro-regions employed for HR detection, the better is the algorithm performance. We also noticed that most of the micro-regions are rejected during point tracking, especially for blocks of higher duration. Therefore, in a future work, special attention should be given to the optical flow algorithm.

The knowledge that, for steady videos, the forehead and the checks are the best regions on the face for HR estimation could be exploited to ameliorate the ROI, improving the performance for HR-FD and/or HR-MRT. For videos with movement it was observed that the forehead is the best region for HR estimation, while the cheeks are rejected most of the time due to motion artifacts.

We could also improve the performance by pre-processing the traces obtained in each time block to reduce the discontinuities artifacts introduced on the transition from one block to another. This could be done by modifying the normalization step imposing the continuity of the DC level and amplitude.

REFERENCES

[1] JAMBUKIA, S.; DABHI, V.; PRAJAPATI, H. Classification of ecg signals using machine learning techniques: A survey. In: *2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)*. [S.l.: s.n.], 2015. p. 714 – 721.

[2] TANTAWI, M. et al. On the use of the electrocardiogram for biometrie authentication. In: *Informatics and Systems (INFOS), 2012 8th International Conference on*. [S.l.: s.n.], 2012. p. BIO– (48 – 54).

[3] SANGURMATH, S.; DAIMIWAL, N. Application of photoplethysmography in blood flow measurement. In: *Industrial Instrumentation and Control (ICIC), 2015 International Conference on*. [S.l.: s.n.], 2015. p. 929 – 933.

[4] WUKITSCH, M. W. Pulse oximetry: Historical review and ohmeda functional analysis. *International journal of clinical monitoring and computing*, Springer Netherlands, v. 4, n. 3, p. 161 – 166, 1987. ISSN 0167-9945.

[5] CAPDEVILA, L. et al. Hrv based health amp and sport markers using video from the face. In: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. [S.l.: s.n.], 2012. p. 5646 – 5649. ISSN 1557-170X.

[6] BAL, U. Non-contact estimation of heart rate and oxygen saturation using ambient light. *Biomed. Opt. Express*, OSA, v. 6, n. 1, p. 86 – 97, January 2015.

[7] COUDERC, J.-P. et al. Pulse harmonic strength of facial video signal for the detection of atrial fibrillation. In: *Computing in Cardiology Conference (CinC), 2014*. [S.l.: s.n.], 2014. p. 661 – 664. ISSN 2325-8861.

[8] LI, X. et al. Remote heart rate measurement from face videos under realistic situations. In: *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014.

[9] XU, S.; SUN, L.; ROHDE, G. K. Robust efficient estimation of heart rate pulse from video. *Biomed. Opt. Express*, OSA, v. 5, n. 4, p. 1124 – 1135, April 2014.

[10] YU, Y.-P. et al. Video-based heart rate measurement using short-time fourier transform. In: *Intelligent Signal Processing and Communications Systems (ISPACS), 2013 International Symposium on*. [S.l.: s.n.], 2013. p. 704 – 707.

[11] PURSCHE, T.; KRAJEWSKI, J.; MOELLER, R. Video-based heart rate measurement from human faces. *Electronics (ICCE)*, p. 544 – 545, 2012.

[12] BOLKHOVSKY, J.; SCULLY, C.; CHON, K. Statistical analysis of heart rate and heart rate variability monitoring through the use of smart phone cameras. In: *Engineering in Medicine*

and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. [S.l.: s.n.], 2012. p. 1610 – 1613. ISSN 1557-170X.

[13] POH, M.-Z.; MCDUFF, D. J.; PICARD, R. W. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics express*, v. 18, n. 10, p. 10762 – 10774, 2010. ISSN 1094-4087.

[14] POH, M.-Z.; MCDUFF, D.; PICARD, R. Advancements in noncontact, multiparameter physiological measurements using a webcam. *Biomedical Engineering, IEEE Transactions on*, v. 58, n. 1, p. 7–11, January 2011. ISSN 0018-9294.

[15] VERKRUYSSE, W.; SVAASAND, L. O.; NELSON, J. S. Remote plethysmographic imaging using ambient light. *Opt. Express*, OSA, v. 16, n. 26, p. 21434 – 21445, December 2008.

[16] KWON, S.; KIM, H.; PARK, K. S. Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, p. 2174 – 2177, 2012. ISSN 1557170X.

[17] ALIAN, A. A.; SHELLEY, K. H. Photoplethysmography. *Best Practice & Research Clinical Anaesthesiology*, v. 28, n. 4, p. 395 – 406, 2014. ISSN 1521-6896. Hemodynamic Monitoring Devices.

[18] JAYADEVAPPA, B. M. et al. Design and development of electro-optical system for acquisition of ppg signals for the assessment of cardiovascular system. *International Journal of Research in Engineering and Technology*, v. 3, p. 520 – 525, June 2014.

[19] ZIJLSTRA, W. G.; BUURSMA, A.; ROEST, W. P. Meeuwsen-van der. Absorption spectra of human fetal and adult oxyhemoglobin, de-oxyhemoglobin, carboxyhemoglobin, and methemoglobin. *Clinical Chemistry*, v. 37(9), p. 1633 – 1638, 1991.

[20] ESTEPP, J.; BLACKFORD, E.; MEIER, C. Recovering pulse rate during motion artifact with a multi-imager array for non-contact imaging photoplethysmography. In: *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. [S.l.: s.n.], 2014. p. 1462 – 1469.

[21] ALLEN, J. Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, v. 28, p. R1 – R39, 2007.

[22] HERTZMAN, A. B. Photoelectric plethysmography of the fingers and toes in man. *Proceedings of the Society for Experimental Biology and Medicine*, v. 37, p. 529 – 534, 1937.

[23] HERTZMAN, A. B.; SPIELMAN, C. Observations on the finger volume pulse recorded photoelectrically. *American Journal of Physiology*, v. 119, p. 334 – 335, 1937.

[24] TRAFFORD, J. de; LAFFERTY, K. What does photoplethysmography measure? *Medical & Biological Engineering & Computing*, v. 22, p. 479 – 480, 1984.

[25] PELEGRIS, P. et al. A novel method to detect heart beat rate using a mobile phone. In: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. [S.l.: s.n.], 2010. p. 5488 – 5491. ISSN 1557-170X.

[26] PAL, A. et al. A robust heart rate detection using smart-phone video. In: *Proceedings of the 3rd ACM MobiHoc Workshop on Pervasive Wireless Healthcare*. New York, NY, USA: ACM, 2013. (MobileHealth 13), p. 43 – 48. ISBN 978-1-4503-2207-2.

[27] WU, H.-Y. et al. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, v. 31, n. 4, p. 651 – 658, 2012.

[28] SZELISKI, R. *Computer Vision: Algorithms and Applications*. [S.l.]: Springer, 2011.

[29] WADHWA, N. et al. Phase-based video motion processing. *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, v. 32, n. 4, 2013.

[30] RAJESHWARI, J.; KARIBASAPPA, K.; GOPALKRISHNA, M. Survey on skin based face detection on different illumination, poses and occlusion. In: *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*. [S.l.: s.n.], 2014. p. 728 – 733.

[31] LIU, Y. et al. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, v. 40, n. 1, p. 262 – 282, 2007. ISSN 0031-3203.

[32] SANTOS, C.; SOUTO, E.; SANTOS, E. dos. Andimage: An adaptive architecture for nude detection in image. In: *Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on*. [S.l.: s.n.], 2015. p. 1 – 6.

[33] KONG, S. G. et al. Recent advances in visual and infrared face recognition—a review. *Computer Vision and Image Understanding*, v. 97, n. 1, p. 103 – 135, 2005. ISSN 1077-3142.

[34] SOCOLINSKY, D. A.; SELINGER, A.; NEUHEISEL, J. D. Face recognition with visible and thermal infrared imagery. *Computer Vision and Image Understanding*, v. 91, n. 1–2, p. 72 – 114, 2003. ISSN 1077-3142. Special Issue on Face Recognition.

[35] KAKUMANU, P.; MAKROGIANNIS, S.; BOURBAKIS, N. A survey of skin-color modeling and detection methods. *Pattern Recognition*, v. 40, n. 3, p. 1106 – 1122, 2007. ISSN 0031-3203.

[36] VEZHNEVETS, V.; SAZONOV, V.; ANDREEVA, A. A survey on pixel-based skin color detection techniques. In: *IN PROC. GRAPHICON-2003*. [S.l.: s.n.], 2003. p. 85 – 92.

[37] ZARIT, B.; SUPER, B.; QUEK, F. Comparison of five color models in skin pixel classification. In: *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*. [S.l.: s.n.], 1999. p. 58 – 63.

[38] TERRILLON, J.-C. et al. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In:

Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on. [S.l.: s.n.], 2000. p. 54 – 61.

[39] GOMEZ, G. On selecting colour components for skin detection. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on.* [S.l.: s.n.], 2002. v. 2, p. 961 – 964. ISSN 1051-4651.

[40] STERN, H.; EFROS, B. Adaptive color space switching for face tracking in multi-colored lighting environments. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on.* [S.l.: s.n.], 2002. p. 249 – 254.

[41] YANG, J.; LU, W.; WAIBEL, A. Skin-color modeling and adaptation. In: *Proceedings of the Third Asian Conference on Computer Vision-Volume II.* London, UK, UK: Springer-Verlag, 1997. (ACCV 98), p. 687 – 694. ISBN 3-540-63931-4.

[42] GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing.* Third edition. [S.l.]: Prentice Hall, 2007.

[43] BRAND, J.; MASON, J. A comparative assessment of three approaches to pixel-level human skin-detection. In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on.* [S.l.: s.n.], 2000. v. 1, p. 1056 – 1059. ISSN 1051-4651.

[44] JONES, M.; REHG, J. Statistical color models with application to skin detection. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* [S.l.: s.n.], 1999. v. 1, p. 280. ISSN 1063-6919.

[45] CAETANO, T.; BARONE, D. A probabilistic model for the human skin color. In: *Image Analysis and Processing, 2001. Proceedings. 11th International Conference on.* [S.l.: s.n.], 2001. p. 279 – 283.

[46] WARK, T.; SRIDHARAN, S. A syntactic approach to automatic lip feature extraction for speaker identification. In: *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on.* [S.l.: s.n.], 1998. v. 6, p. 3693 – 3696. ISSN 1520-6149.

[47] FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach.* Second edition. [S.l.]: Prentice Hall, 2011.

[48] POYNTON, C. A. *Frequently Asked Questions About Color.* March 1997.

[49] KOVAC, J.; PEER, P.; SOLINA, F. Human skin color clustering for face detection. In: *EUROCON 2003. Computer as a Tool. The IEEE Region 8.* [S.l.: s.n.], 2003. v. 2, p. 144 – 148.

[50] TSEKERIDOU, S.; PITAS, I. Facial feature extraction in frontal views using biometric analogies. In: *European Signal Processing Conference.* [S.l.: s.n.], 1998. p. 315 – 318.

[51] CHAI, D.; NGAN, K. Face segmentation using skin-color map in videophone applications. *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 9, n. 4, p. 551 – 564, June 1999. ISSN 1051-8215.

[52] KHAN, R. et al. Static filtered skin detection. *International Journal of Computer Science Issues*, v. 9, n. 3, p. 257 – 261, March 2012.

[53] YANG, M. H.; AHUJA, N. Gaussian mixture model for human skin color and its application in image and video databases. *Proceedings of SPIE: Conference on Storage and Retrieval for Image and Video Databases*, v. 12, p. 987 – 1003, 2000.

[54] RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Second edition. [S.l.]: Pearson Education, 2003.

[55] AL-MOHAIR, H. K.; MOHAMAD-SALEH, J.; SUANDI, S. A. Human skin color detection: A review on neural network perspective. *International Journal of Innovative Computing, Information and Control*, v. 8, n. 12, p. 8115 – 8131, December 2012.

[56] BALAKRISHNAN, G.; DURAND, F.; GUTTAG, J. Detecting pulse from head motions in video. In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. [S.l.: s.n.], 2013. p. 3430 – 3437. ISSN 1063-6919.

[57] ASTHANA, A. et al. Robust discriminative response map fitting with constrained local models. In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. [S.l.: s.n.], 2013. p. 3444 – 3451. ISSN 1063-6919.

[58] WELCH, P. D. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Audio and Electroacoustics, IEEE Transactions on*, v. 15, n. 2, p. 70 – 73, June 1967. ISSN 0018-9278.

[59] VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.: s.n.], 2001. v. 1, p. I–(511 – 518). ISSN 1063-6919.

[60] LIENHART, R.; MAYDT, J. An extended set of haar-like features for rapid object detection. In: *Image Processing. 2002. Proceedings. 2002 International Conference on*. [S.l.: s.n.], 2002. v. 1, p. I–(900–903). ISSN 1522-4880.

[61] CARDOSO, J.-F. High-order contrasts for independent component analysis. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 11, n. 1, p. 157 – 192, january 1999. ISSN 0899-7667.

[62] HAUSMAN, K.; MERRICK, E. *Pulse oximeter*. [S.l.]: Google Patents, november 1989. US Patent 4,883,353.

[63] WIERINGA, F. P.; MASTIK, F.; STEEN. Contactless multiple wavelength photoplethysmographic imaging: A first step toward spo2 camera technology. *Annals of Biomedical Engineering*, v. 33, n. 8, p. 1034 – 1041, 2005. ISSN 0090-6964.

[64] HUMPHREYS, K.; WARD, T.; MARKHAM, C. Noncontact simultaneous dual wavelength photoplethysmography: A further step toward noncontact pulse oximetry. *Review of Scientific Instruments*, v. 78, n. 4, p. 044304 – (1–6), 2007.

[65] WEISSTEIN, E. *The CRC Concise Encyclopedia of Mathematics*. New York: CRC Press LLC, 2002.

[66] BEUCHER, S.; LANTUéJOUL, C. *Use of Watersheds in Contour Detection*. september 1979. Workshop published.

[67] TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Computer Vision, 1998. Sixth International Conference on*. [S.l.: s.n.], 1998. p. 839 – 846.

[68] SHI, J.; TOMASI, C. Good features to track. In: *Proceedings CVPR 94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 1994. p. 593 – 600. ISSN 1063-6919.

[69] YVES, B. J. Pyramidal implementation of the lucas-kanade feature tracker. *Microsoft Research Labs, Tech. Rep*, 1999.

[70] LAWSON, C. L.; HANSON, R. J. *Solving Least Squares Problems*. Englewood Cliffs, NJ 07632, USA: Prentice-Hall, 1974. 0 p. (Series in Automatic Computation).

[71] SAYOOD, K. *Introduction to Data Compression*. Fourth edition. [S.l.]: Morgan Kaufmann, 2012.

[72] BAYER, B. *Color imaging array*. [S.l.]: Google Patents, july 1976. US Patent 3,971,065.

APPENDIX

I. AFFINE TRANSFORMATION MATRIX

The affine transformation matrix can be given by a combination of simple transformations, as

$$R = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \alpha \begin{bmatrix} (-1)^m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & \alpha_h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha_v & 1 \end{bmatrix}$$

         ↳ vertical shear
      ↳ horizontal shear
   ↳ rotation
↳ mirroring
↳ scaling

$$\tag{I.1}$$

We assume that $det(R) \neq 0$. Therefore we can find the values of m, α, θ, α_h and α_v that represent this transformation.

Scaling and mirroring can be determined directly from the matrix determinant:

$$m = \begin{cases} 1, & \text{if } det(R) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{I.2}$$

$$\alpha = \sqrt{|det(R)|} \tag{I.3}$$

Let us now modify matrix R in such a way that the determinant of the modified matrix is equal to 1, resulting in

$$R' = \begin{bmatrix} a'_{11} & a'_{12} \\ a'_{21} & a'_{22} \end{bmatrix} = \frac{1}{\alpha} \begin{bmatrix} (-1)^m & 0 \\ 0 & 1 \end{bmatrix} R \tag{I.4}$$

From this modified matrix we can compute the other parameters as:

$$\alpha_h = \pm\sqrt{a'^2_{12} + a'^2_{22} - 1} \tag{I.5}$$

$$\alpha_v = \frac{a'_{11}a'_{12} + a'_{21}a'_{22} - \alpha_h}{a'^2_{12} + a'^2_{22}} \tag{I.6}$$

$$cos(\theta) = \frac{a'_{12}\alpha_h + a'_{22}}{1 + \alpha_h^2} \tag{I.7}$$

$$sin(\theta) = \frac{a'_{22}\alpha_h - a'_{12}}{1 + \alpha_h^2} \tag{I.8}$$

Notice that there are two possible solution for α_h, α_v and θ depending on the sign of α_h, but both solutions lead to the same matrix R.

II. RANDOM ESTIMATION OF THE HEART RATE

Consider two independent random variables I and E that represent the input and estimated frequency, respectively. Assume that I is uniform between 60 and 200 and E is uniform between 30 and 240. Their joint probability distribution, $f(i, e)$, is depicted in Figure II.1. As they are independent, $f(i, e)$ is given by the product of the probability distribution function of I and E and is, therefore, constant inside the drawn rectangle and zero outside.

The joint probability distribution is restricted to the constraint:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(i, e) \, di \, de = \int_{60}^{200} \int_{30}^{240} K \, di \, de = 1, \tag{II.1}$$

where K is the constant level of $f(i, e)$ inside the rectangle and is found to be equal to the inverse of the rectangle area given by $(200 - 60)(240 - 30)$.
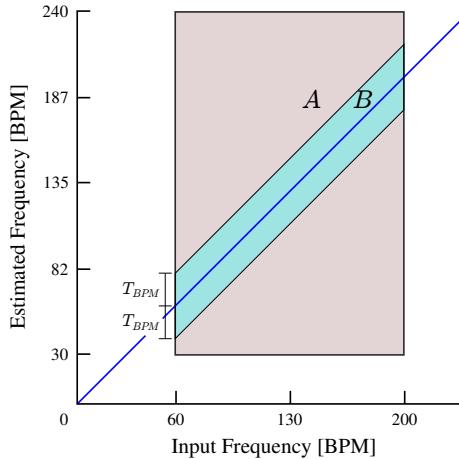

Figure II.1: Joint probability distribution.

The blue line correspond to the points where $I = E$ and the region marked in blue correspond to all points where $|I - E| \leq T_{BPM}$. Thus, $P\{|I - E| \leq T_{BPM}\}$ is given by the integral of $f(i, e)$ inside the blue region and is equivalent to the ratio of the blue region area by the total area of the rectangle.

When $T_{BPM} \leq 30$, the upper and bottom frontiers of $|I - E| \leq T_{BPM}$ do not cross the upper and bottom lines of the rectangle and the area of the blue region is given by $2T_{BPM}(200 - 60)$. Therefore,

$$P\{|I - E| \leq T_{BPM}\} = \frac{2T_{BPM}(200 - 60)}{(200 - 60)(240 - 30)} = \frac{T_{BPM}}{105}, \tag{II.2}$$

which, even though it may be small, it is higher than zero.