Robust pulse waveform extraction from rPPG

Author:
ing. K. (Koen) de Laat

Supervisor:
prof.dr.ir. G. (Gerard) de Haan
PhD E. (Erik) Bresch

A thesis submitted in fulfillment of the requirements
for the degree of Embedded Systems

in the

Electronic Systems Group
Mathematics and Computer Science - Electrical Engineering

August 2013
Remote photoplethysmography (rPPG) enables contactless monitoring of the blood volume pulse (BVP) using a regular camera. Recent research has focused on motion robustness and selecting a high quality site of skin. This thesis is about finding a clean pulse waveform with rPPG. We present a method to extract the pulse waveform by using a temporal average over multiple periods. We also present a method that uses the pulse waveform as a selection criterion to improve the signal to noise ratio (SNR) of the BVP signal. Multiple experiments are performed to test the system, one of them is a test of the complete system based on generated, synthetic signals. Also, differences in pulse waveform over individuals are examined. Furthermore the different methods (existing and proposed) are compared on multiple recordings. The experiments show an average increase in SNR of 1.55 dB ($\sigma = 1.09, p = 0.003$) when using our proposed method with respect to an inverse variance weighted method.
Contents

Abstract ii

List of Figures v

List of Tables vii

1 Introduction 1
  1.1 Photoplethysmography . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
    1.1.1 Remote Photoplethysmography . . . . . . . . . . . . . . . . . . . . . 3
  1.2 Pulse waveform . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
    1.2.1 Pulse waveform and rPPG . . . . . . . . . . . . . . . . . . . . . . . . 4
  1.3 Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2 Data acquisition and pre-processing 7
  2.1 Camera recording . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2.2 Extract color signals . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2.3 Color to blood volume pulse signal . . . . . . . . . . . . . . . . . . . . . 9
    2.3.1 Inverse variance weighted sum . . . . . . . . . . . . . . . . . . . . . 9
    2.3.2 Alpha-trimmed mean . . . . . . . . . . . . . . . . . . . . . . . . . . 10
    2.3.3 Overlap-add . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  2.4 Performance metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
    2.4.1 SNR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
    2.4.2 Consistency of the pulse . . . . . . . . . . . . . . . . . . . . . . . . . 12

3 Temporal processing 13
  3.1 Normalization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  3.2 Heart-rate detection using the spectral domain . . . . . . . . . . . . . . . 15
  3.3 Interval detection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
    3.3.1 Peak detection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
    3.3.2 Determine interval . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
      3.3.2.1 Peak to peak . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      3.3.2.2 Zero-crossings . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
  3.4 Normalize pulses and average . . . . . . . . . . . . . . . . . . . . . . . . 18

4 Spatial processing 19
  4.1 Pulse waveform correlation . . . . . . . . . . . . . . . . . . . . . . . . . 19
  4.2 Normalization and trimmed-mean . . . . . . . . . . . . . . . . . . . . . . 20
4.3 Determine amplitude, frequency and phase .................................. 21
  4.3.1 Frequency adjustment heuristics ........................................ 21
  4.3.2 Increased window ......................................................... 22
4.4 Synthetic reference ............................................................. 22
  4.4.1 Pulse shape ............................................................... 23
4.5 Correlate and select signals .................................................... 23

5 Pulse waveform feedback .......................................................... 27
  5.1 Length of a single run ......................................................... 28
  5.2 Introduced difficulties ........................................................ 29
    5.2.1 Normalization ............................................................ 29
    5.2.2 FFT ................................................................. 29
    5.2.3 Peak detection ........................................................ 29
    5.2.4 Interval detection ..................................................... 30
    5.2.5 Pulse waveform ........................................................ 30

6 Algorithm validation ............................................................... 31
  6.1 Pulse waveform ............................................................... 31
    6.1.1 Pre-defined shape ..................................................... 31
    6.1.2 Iterative feedback ................................................... 33
    6.1.3 Complete feedback ................................................... 34
    6.1.4 Volunteer related pulse waveform .................................. 35
  6.2 Robustness to noise and interfering signals ................................ 36
    6.2.1 Noise ................................................................. 36
    6.2.2 Noise and interfering frequencies .................................. 37
  6.3 SNR improvements ............................................................. 39

7 Conclusions and future work .................................................... 43
  7.1 Future work ................................................................. 44

Bibliography .............................................................................. 45
## List of Figures

1.1 General PPG principal ........................................ 2  
1.2 Cross section of the human skin .............................. 2  
1.3 Pulse-oximeter ................................................ 3  
1.4 PPG principal .................................................. 4  
1.5 Block diagram complete processing ......................... 6  
2.1 Screenshot of ”make-traces” ................................ 8  
2.2 Illustration of overlap-add procedure ....................... 11  
2.3 SNR weights template ......................................... 12  
3.1 Block diagram temporal processing ......................... 13  
3.2 Normalization in the temporal processing .................. 15  
3.3 Spectral domain for detecting the heart-rate ............... 16  
3.4 Peak and interval detection on a rPPG signal ............... 18  
4.1 Block diagram original spatial processing .................. 19  
4.2 Block diagram proposed spatial processing ................. 20  
4.3 Examples of extracted BVP traces ........................... 21  
4.4 Linear and Cosine shaped pulse ............................. 23  
4.5 Examples of extracted BVP traces and resulting signal ... 24  
5.1 Block diagram complete processing including feedback .... 27  
6.1 Results of linear and cosine pulse shapes ................... 32  
6.2 Results of iterative feedback pulse waveform ............... 33  
6.3 Results of complete feedback pulse waveform .............. 34  
6.4 Pulse waveforms of volunteers ............................... 35  
6.5 Robustness to noise ......................................... 38  
6.6 Robustness to noise and interference ....................... 39  
6.7 SNR of the different methods ............................... 40
List of Tables

6.1 SNR of the different methods ................................................. 40
Heart-rate (HR) is the number of heartbeats per unit of time, often expressed in beats per minute (BPM). The heart-rate can vary over time due to need of change in absorbing oxygen or excrete carbon dioxide. These changes can be causes by different physical activities like exercise or sleep. The heart-rate is used by medical professionals to assist in determining the patient medical conditions. Heart-rate is also used by athletics who use this information to improve their training and performance.

Whenever the heart beats, it pumps blood through the circulatory system, the system of blood vessels. Due to this blood flow a blood pressure pulse travels from the heart to other body parts. This is typically referred as a Blood Volume Pulse (BVP). The heart-rate can be detected if the BVP signal can be detected anywhere on the body. This method of detecting the heart-rate based on the BVP signal is called Plethysmography (‘Plethysmos’ means increase in Greek).

1.1 Photoplethysmography

Photoplethysmography (PPG) is an optical method to detect the BVP signal. This method is introduced in 1930s [1] and is a simple, non-invasive, low-cost method and highly popular for these reasons. It detects the BVP signal using a light source and detector and is based on small variations in the light intensity of the skin due to BVP changes in the microvascular bed of tissue. A general set-up of capturing a PPG signal is shown in Figure 1.1.

When light interacts with biological tissue a complex interaction takes place including the optical processes of scattering, absorption, reflection, transmission and fluorescence. Bones, tissues and blood absorb the dominant part of the light, where blood absorbs
more light than the surrounding tissue. The important factor that affects the amount of absorbed light by blood are the blood volume, blood vessel wall movement and the orientation of red blood cells [3]. The skin, which is highly popular for PPG analysis, is divided into three layers viz. Epidermis, Dermis and Subcutaneous (also called Hypodermis), see Figure 1.2. The skin is richly perfused and has the congregation of multiple veins, also called venous plexus, in Dermis and Subcutaneous. With each heart-beat the blood is pumped to the periphery. In the subcutaneous tissue the pressure pulses are somewhat dampened but still enough to distend the arteries and arterioles. It is relatively easy to detect the pulsating component of the cardiac cycle in the skin. Often a pulse-oximeter is used to obtain a PPG signal. In this pulse-oximeter a LED and photodiode are used to illuminate the skin and measuring the amount of reflected or transmitted light. An example of a pulse-oximeter is shown in Figure 1.3.

Due to the historical emphasis of PPG on pulse oxymetry and the associated need to sample relatively deep (e.g. 1 mm) veins and arteries, the visible spectrum has often
been ignored as a light source for PPG. Originally a dedicated light source was used, typically red or infrared (IR), with a deeper penetration depth of the skin. Earlier work on using non-red, visible light sources for PPG were always using contact probes [6, 7, 8]. The ambient visible light is often considered a source of noise [9, 10] when using IR light sources and detectors sensitive for IR and visible light.

1.1.1 Remote Photoplethysmography

Relatively recent work shows us that remote, non-contact, pulse oxymetry and PPG imaging is possible using SpO(2) camera technology and three different wavelengths LED [11, 12]. More recent work shows successful results with remote-PPG (rPPG) using a regular color camera in ambient light conditions [13, 14, 15, 16]. rPPG is a relevant topic as it can be used in cases where contact has to be prevented because of extreme sensitivity, e.g. neonates, skin-damage, or when an increased unobtrusiveness is required/desired, e.g. surveillance or fitness.

The main problem with rPPG is that the signal quality (measured in SNR) is quite low compared to contact PPG where a dedicated source of light is used with an unmoving skin patch. Recent research is done to improve the rPPG signal quality under varying light conditions and motion robustness [13, 14, 17]. Almost all motion robust rPPG techniques use the fact that the optical absorption of the human skin is dependent on the wavelength of the light used, whereas motion of the skin relative to the camera is not dependent of the wavelength. An illustration of this principle is shown in Figure 1.4.
1.2 Pulse waveform

Next to the heart-rate and respiration-rate also other information can be extracted from the PPG signal. The waveform of an individual pulse of one period contains information which is valuable for medical applications. For example the pulse transit time can be examined with an ECG and PPG signal [18]. The pulse waveform can also give information about the vascular condition or autonomic nervous system [19, 20]. Spigulis et al. [20] show a method to extract the pulse waveform from a contact PPG sensor. They also tested their method on diabetes patients and show that detection of diabetes is possible with the waveform of a PPG signal. Sukor et al. [21] use the pulse waveform to distinguish between good and poor pulses to improve the pulse oximetry quality.

1.2.1 Pulse waveform and rPPG

To the best of our knowledge no previous work is done on extracting information from a pulse waveform of a rPPG signal. We think that it would be useful to have a possibility to extract similar information from the rPPG pulse waveform as available from the PPG pulse waveform. Especially this can be interesting for cases where unobtrusiveness is required, like in long-term surveillance of patients. It can also be useful for monitoring neonates to possibly detect arterial disease in an early stage without damaging the skin. The problem with rPPG and waveform analysis is that a single pulse waveform from rPPG contains too much noise due to interference of other light sources and movement.
of the patient. As the waveform contains a lot of noise, it is not possible to extract information out of it in a simple way.

1.3 Objectives

We want to develop a method to extract the pulse waveform based on a rPPG signal. We will try to use methods similar to [20, 21] to identify individual pulses and averaging those to improve the waveform quality. We will extend the existing methods so they are capable of detecting pulse intervals on noisy signals with low sample rate in comparison with PPG signals. We will also try to improve the rPPG signal quality by using information stored in the pulse waveform with relation to the state of the art methods looking at the metrics as defined in Section 2.4. Our first try will be improving the signal quality based on a general template pulse waveform. After accomplishing that, we will try to use a feedback mechanism which uses the pulse waveform from already processed data to improve the signal quality of data ready for processing.

Figure 1.5 gives an overview of the steps of the complete processing from camera recording until pulse waveform. In Chapter 2 we will explain the steps used to extract the BVP signal from the camera recording. It also explains the current state of the art methods used for the spatial processing. Furthermore it contains the performance metrics used in the experiments. Chapter 3 explains the temporal processing where individual pulses are segmented and combined to extract an average pulse waveform. The spatial processing is explained in Chapter 4. In the spatial processing we use the information of the pulse waveform as a selection criterion to improve the BVP signal. In Chapter 5 we extend the diagram of Figure 1.5 with a feedback system as illustrated in Figure 5.1. In this chapter we discuss how we use the pulse waveform and how we solved the difficulties introduced by the feedback mechanism. Chapter 6 contains the experiments that are performed to validate and test our proposed methods and shows the results of these experiments. Finally we draw our conclusions in Chapter 7 and propose possible future work.
Figure 1.5: A block diagram describing the building blocks of the complete processing.
Chapter 2

Data acquisition and pre-processing

The data that we use for the rPPG processing is originated from a digital camera as explained in Section 1.1.1. A digital camera produces a sequence of images whereas we will do our processing on BVP traces. The conversion of an image sequence to BVP traces is outside the scope of this thesis however is explained in this chapter and called pre-processing. Also the existing methods for the spatial processing are explained in this chapter.

2.1 Camera recording

We used a 768 × 572 pixels, 8 bit, global shutter RGB CCD camera (type UI-222xSE-C of IDS GmbH) operated at 20 pictures per second. The recording is stored in an uncompressed format. In almost all experiments we try to minimize any movements and to maximize the region of interest (ROI). This is achieved by using a support while recording a hand or to sit still while recording a face. The ROI is in our case a patch of skin, so the hand or face.

2.2 Extract color signals

To extract the BVP signal from the image sequence we first extract traces of color signals from the sequence. This is done by a tool developed by Philips called “make-traces” which takes an image sequence as input and produces an output file containing the average values of the colors of each block. The tool needs a region of interest (ROI) to
be able to function correctly. It puts a grid on top of the image sequence and compares the color of each block in the grid with the color inside and outside the ROI. Using this comparison the tool determines whether this block is an interesting block (containing skin and possible a BVP signal) or not (background, hair, ...). The grid is motion compensated using a global motion estimator based on the ROI. A screenshot of this process is show in Figure 2.1. This selection of blocks is done every 16 frames and if a block is selected, the color trace is calculated for the next 32 frames. This leads to an overlap of traces with 16 frames with the previous trace and 16 frames with the next trace. These partly overlapping traces are used in an overlap-add fashion using Hann windowing on individual traces in the conversion from BVP traces to a single BVP signal as explained in Section 2.3.3.

Calculating a color trace is done by averaging the color value over all pixels in a block per frame. This leads to three values per block per frame for a single trace. For a trace \( T \) and color \( C \in \{R, G, B\} \) the color trace value of frame \( i \) can be calculated as:

\[
BV_{T,C,i} = \frac{1}{N} \sum_{j=1}^{N} PV_{C,i,j}
\]

(2.1)

where \( N \) is the number of pixels in a block and \( PV_{C,i,j} \) is the value of color \( C \) in pixel \( j \) of this block in frame \( i \).
2.3 Color to blood volume pulse signal

The conversion from color traces to blood volume pulse traces is done using the method proposed by de Haan and Jeanne [17]. The algorithm used is the one referred as $X_s\min_Y$. The idea of this method is to use color difference signals where the variations due to blood volume changes in the skin will likely be different, while motion affects both color difference signals identically. These color difference signals are composed in such a way that they can work correct regardless the color of the illumination. The pulse signal $S$ can be calculated according to de Haan and Jeanne [17] using:

$$S = X_f - \alpha Y_f$$  \hspace{1cm} (2.2)

with

$$\alpha = \frac{\sigma(X_f)}{\sigma(Y_f)}$$  \hspace{1cm} (2.3)

where $\sigma(x)$ is the standard deviation of $x$ and $X_f$ and $Y_f$ are the band-passed filtered versions of $X_s$ and $Y_s$ with

$$X_s = 3R_n - 2G_n$$  \hspace{1cm} (2.4)

$$Y_s = 1.5R_n + G_n - 1.5B_n$$  \hspace{1cm} (2.5)

where $C_n$ is the individual color channel $C \in \{R,G,B\}$ divided by the mean of this channel.

Using this method we can convert a single color trace, consisting of 32x3 values, into a BVP trace consisting of only 32 values. As described in Section 2.2 the "make-traces" tool selected multiple blocks containing a possible BVP signal, leading to multiple simultaneous traces. To combine these traces into a single BVP signal, multiple existing solutions exists and we call it spatial processing.

2.3.1 Inverse variance weighted sum

One of the options for spatial processing is to use an inverse variance weighting sum, where each individual trace is weighted in inverse proportion to its variance and the result is the sum of all the weighted traces, so the combined BVP signal $S_t$ of traces $S_{\{1..M\},t}$ is calculated as:

$$S_t = \sum_{tr=1}^{M} \frac{S_{tr,t}}{\sigma^2(S_{tr})}$$  \hspace{1cm} (2.6)

The idea behind this inverse variance weighted method is that traces that have a high variance are likely to contain a lot of noise. The down-side of this method is that blocks
containing a strong BVP signal also have a high variance and get a low weight, which is an unwanted situation.

2.3.2 Alpha-trimmed mean

Another existing method is to use an alpha-trimmed mean for combining the traces. The idea behind this method is that all traces that contain a BVP signal look similar and the other traces, containing only noise, are considered outliers. To calculate the alpha-trimmed mean we first need to sort the data. So for the relative point in time \( t \in \{1..32\} \) and for all simultaneous traces \( tr \in \{1..M\} \) the data \( S_{tr,t} \) is sorted in such a way that

\[
\forall t \forall tr \in \{1..M-1\} \quad S_{tr,t} \leq S_{tr+1,t}
\]  

(2.7)

The alpha-trimmed mean is calculated as:

\[
S_t = \frac{1}{(1-\alpha)M} \sum_{tr=\frac{1}{2}M+1}^{(1-\alpha)M} S_{tr,t}
\]

(2.8)

Where \( 0 \leq \alpha < 1 \) and is set to 0.5 in the current implementation.

For both methods the resulting \( S \) is normalized so that the resulting signal has a standard deviation of one using:

\[
S_{nt} = \frac{S_t}{\sigma(S)}
\]

(2.9)

2.3.3 Overlap-add

To combine the individual traces into a single BVP single we use an overlap-add method. This method is the same method as used by de Haan and Jeanne [17]. They show that there is an advantage in separately optimizing partially overlapping intervals and gluing the resulting pieces together using Hann windowing on individual intervals. For interval number \( N = \lfloor 2t/\text{interval} \rfloor \) and trace \( S_{N,t} \), composed of the frames \( t \) within the interval \( N \), we can calculate the resulting signal as:

\[
S_t = \sum_N wh_{N,t} \times S_{N,t}
\]

(2.10)

where \( wh_{N,t} \) is the Hann windowing function centered in interval \( N \) and zero outside the interval:

\[
wh_{N,t} = 0.5 - 0.5 \cos(2\pi t/\text{interval})
\]

(2.11)
Chapter 2. Data acquisition and pre-processing

2.4 Performance metrics

To evaluate and compare results from different methods we need some performance metrics. We used two metrics, one is the signal to noise ratio (SNR) and the other is the mean Euclidean distance of different pulses.

2.4.1 SNR

The signal to noise ratio (SNR) is a metric to compare the level of a desired signal to the level of background noise or unwanted signals. It is defined as the ratio of signal power to the noise power and expressed in decibels. We use the SNR metric in the spectral domain, where the heart-rate frequency and the first harmonic are considered as signal and the other frequencies in the range 1-240 beats per minute (BPM) are considered noise. Due to the variations in the heart-rate there is not exactly one heart-rate but a small region of heart-rates. That is why we use the average power of a region of 10 BPM and the corresponding first harmonic region of 20 BPM as signal. To calculate the SNR we make use of the weighting function $w(f, hr)$ as in Equation 2.12 and shown in
Chapter 2. Data acquisition and pre-processing

Figure 2.3: Template used for the SNR weights. Figure shows an example for a heart-rate of 80 BPM.

Figure 2.3.

\[ w(f, hr) = \begin{cases} 
1 & \text{if } (hr - 5) < f < (hr + 5) \\
1 & \text{if } (2hr - 10) < f < (2hr + 10) \\
0 & \text{otherwise} 
\end{cases} \]  
(2.12)

The actual SNR in decibel is calculated as:

\[
SNR_{dB} = 10 \log_{10} \left( \frac{\sum_{f=1}^{240} (y(f) \times w(f, hr))^2}{\sum_{f=1}^{240} (y(f) \times (1 - w(f, hr)))^2} \right) 
\]  
(2.13)

2.4.2 Consistency of the pulse

The metric consistency is used in the temporal processing and describes the variation between the average pulse waveform and the individual pulses used to calculate this average. We use the average Euclidean distance between the average pulse waveform and the individual pulses. The Euclidean distance between the average waveform \( W_t \) and pulse \( P_{i,t} \) with a length of \( N \) is defined as:

\[
ed_i = \sqrt{\sum_{t=1}^{N} (P_{i,t} - W_t)^2} \]  
(2.14)

We use the average Euclidean distance over all \( M \) pulses used in the trimmed-mean as our metric resulting in:

\[
\text{consistency} = \frac{1}{M} \sum_{i=1}^{M} ed_i \]  
(2.15)
Chapter 3

Temporal processing

As described in Section 1.2 there are several applications where the pulse waveform of the (r)PPG signal can be helpful. As mentioned in Section 1.2.1 the pulse waveform of a single pulse in the rPPG signal contains too much variations and noise to make it useful for medical applications. To overcome this problem we developed a method similar to Sukor et al. [21] to improve the pulse waveform by normalizing and averaging multiple pulses. The details of this method are explained in this chapter. An overview of the steps inside the temporal processing is illustrated in Figure 3.1.

Figure 3.1: A block diagram describing the steps inside the temporal processing.
3.1 Normalization

The amplitude of the rPPG signal is affected by several external factors, for example varying light conditions. To make the averaging of multiple pulses simpler the rPPG signal is first normalized before processed. This is done in such a way that the normalization factor can vary over time. Using a constant factor, or a constant factor per period, cannot correct for changing amplitude during a single period. That is why we use a special normalization procedure, which is based on the Empirical Mode Decomposition (EMD) [22]. This procedure detects the peaks and troughs of the input signal and uses this information to normalize the input data.

Here a new problem introduces as the input is recorded at 20 frames per second and with a heart-rate of 60 BPM only 20 samples per period are available, where with a higher heart-rate even less samples per period are available. This can cause a significant error in time between the real (continuous signal) peak and the detected (sampled signal) peak. That is why the complete input signal is up-sampled with a factor of 10, resulting in a sample frequency of 200 Hz. Before this up-sampling a median filter with a kernel size of three is applied to the input signal to filter out the noise spikes. If this median filter was not used these spikes would be widened in the up-sampling and harder to remove further in the process.

The peaks and troughs for the normalization are detected in the up-sampled signal, with an extra median filter with kernel size of five. This extra filter is used to smooth the signal and avoids detecting false peaks and troughs. As there is no information available of the heart-rate at this moment the distance between peaks and troughs is unknown. So the only way to detect peaks (and troughs) in this signal is using the derivative, the minimal value difference between a peak and trough and the fact that peaks and troughs alternate. The threshold for the minimal value difference between a peak and trough is set to \(0.8 \times \sigma\) (filtered signal). The peaks and troughs are used to calculate two splines. One spline that goes through the peaks and one that goes through the troughs. Using these splines \(Sp\) and \(St\) the input signal is normalized by the following equation where \(Su\) is the up-sampled signal.

\[
Sn_t = \frac{Su_t - 0.5 \times (Sp_t + St_t)}{Sp_t - St_t}
\]  

Using this formula the normalization can be different for every \(t\) but the resulting signal will vary around 0 with peaks at 0.5 and troughs at \(-0.5\). This does not implicate that the mean of the signal equals zero because the balance between the positive and negative
values does not have to be equal, in fact it is not with a pulse-shape of the heartbeat. An example of this normalization process is shown in Figure 3.2.

### 3.2 Heart-rate detection using the spectral domain

Detecting the heart-rate is an important aspect for our method. The individual pulses are determined by their peaks and troughs, where the heart-rate is helpful in detecting these peaks and troughs. The heart-rate is also used in to filter out detected pulses with a non-matching length.

The heart-rate is determined in the spectral domain of the signal. This spectral domain is calculated with a Fast Fourier Transformation (FFT) with 8192 bins resulting in a precision of 0.024 Hz or 1.46 BPM. The heart-rate is determined by the bin with the highest value in the range between 30 and 240 BPM. An example of the spectral domain of a recording of a hand is shown in Figure 3.3.
3.3 Interval detection

To determine the intervals of individual pulses the already gathered information from the data is used. We know that the data is normalized, so the value difference between a peak and trough is known. Also the heart-rate is known so we can determine the time difference between two consecutive peaks or troughs.

3.3.1 Peak detection

The peak detection is based on the derivative of the signal together with the already gathered information. As the signal is normalized the peaks should have a value of 0.5 and the troughs −0.5. To allow some variation, the minimal value difference between a peak and trough is set to 0.8. Based on the heart-rate we set the minimal distance between a peak and trough at $0.3 \times \frac{10f_s}{f_{heart}}$ samples. This allows some variation in heart-rate and also difference between the rise and fall time.

3.3.2 Determine interval

To detect the pulse intervals a decision has to be made about what identifies the start of an interval. This can be either a peak, a trough or one of the two zero-crossings with
either a positive or negative derivative. This choice will also affect the start point of the resulting pulse-shape, but can be modified later by a circular shift of the pulse-shape to get another starting point. That is why the starting point of the resulting pulse-shape is not used as a criterion for the interval detection.

### 3.3.2.1 Peak to peak

An easy implementable starting point is a peak, but this has some drawbacks (the same holds for a trough). The start and end of an interval can be determined by a peak \( p_i \) and the next peak \( p_{i+1} \). One disadvantage is the influence of noise resulting in a small error in the peak detection and so the starting (and ending) point cannot be precise. This leads to a non-uniform shape length and also combining these signals can lead to some small errors at the start and end of the combined pulse-shape. Another disadvantage is the value of the peaks. Ideally these are all at 0.5 but due to some variance in the signal this can vary in a small amount. This can lead to problems as the value of the start of a pulse does not match the value of the end of that pulse, \( S_N(p_i) \neq S_N(p_{i+1}) \), resulting in a discontinuous signal.

### 3.3.2.2 Zero-crossings

The other possibility is to use the zero-crossings of the signal. In each pulse two zero-crossings are available, so two possible starting points are available. We think the best detectable zero-crossing is the one on the steepest edge, because noise has the least affect on this edge. We tested the two possible zero-crossings and based on the recordings seen so far our expectation seems correct. The problem with zero-crossings is the relation with the DC portion of the signal, that is why we take the middle between the peak and trough as zero-crossing for each pulse, the value of this zero-crossing if further referred as 'zero’. Since our signal is a sampled signal, finding the sample with the exact value 'zero’ is not always possible. So we need to select a sample with a value close to ’zero’. We select the sample with smallest absolute difference with ’zero’.

In Figure 3.4 is an example shown of the interval detection mechanism where the peaks and troughs are marked with a red circle. The begin and end points of the interval are marked with a black \( \times \) and connected with a black line. The figure shows that the peak-detection is not always perfect as it did not select the exact location of the third trough from the left. It is also visible that the 'zero’ point can vary over different periods.
Figure 3.4: The blue curve is the extracted, normalized rPPG signal. The red circles in this figure indicate the detected peaks and troughs where the black × indicates the start and end points of the intervals. As we look at the third trough from the left we see that the peak detection is not always capable of selecting the correct location. This has to do with the noise in this particular trough. This rPPG signal is extracted from a recording of a hand.

3.4 Normalize pulses and average

After the interval detection we have the begin and end markers of each pulse. With these markers we can extract the individual BVP pulses from the normalized signal. To average the individual pulses we first need to normalize each pulse so that the length of all pulses are similar. This is done by re-sampling each pulse in such a way that each pulse consists of exactly 250 samples. To further normalize the data we change the DC offset of each pulse such that the first sample of a pulse is at value 0.

With these normalized pulses we can average the pulses. This is done using a trimmed mean to remove outliers. These outliers can have its origin in noise in the BVP signal, errors in detecting intervals or a physiologic origin. We use a trimmed mean of 50% of the data.
Chapter 4

Spatial processing

One of the steps in constructing the BVP signal is the spatial processing. This step combines the individual BVP traces created by the "make-traces" and "ExtractXY" tools into a single BVP signal. The two existing options are explained in Section 2.3.1 and Section 2.3.2 and illustrated in Figure 4.1.

4.1 Pulse waveform correlation

The problem with the original methods is that they try to distinguish between blocks (traces) that contain a BVP signal and those that contain noise. But there is no clear definition for these methods how a signal should look like. That is where our new proposed spatial processing is based on. Within our proposed method we try to guide the algorithm to select only those traces that look like a BVP signal that we expect.

Figure 4.1: Block diagrams describing the two original spatial processing alternatives.
We do this by using the alpha-trimmed mean as a reference for the current BVP signal. This reference is similar to one of the original methods and suffers from the same problems. That is why we extract the amplitude \(A\), frequency \(f\) and phase \(\phi\) from this reference and use this to generate a synthetic reference. This reference is based on \(A, f, \phi\) and on a previously stored pulse shape. Next we calculate the correlation between each trace and the synthetic reference signal and based on these correlation coefficients we select the best traces. As output we take the mean of the selected traces and combine this with the signal calculated from previous traces using an overlap-add (see Section 2.3.3 for the overlap-add procedure). Each individual step is explained in detail below. An overview of the steps is visible in Figure 4.2.

### 4.2 Normalization and trimmed-mean

As first step all traces from a single set are normalized using:

\[
S_{n_{tr,t}} = \frac{S_{tr,t} - \mu(S_{tr})}{\sigma(S_{tr})}
\]  

(4.1)

This normalization step makes sure that every trace that does contain a BVP signal has the same amplitude. Figure 4.3 shows 8 examples of normalized traces of the same recording at the same point in time. To come to a first reference signal the trimmed mean method as explained in Section 2.3.2 is used on the normalized signals. This results in a reference signal of 32 samples containing a noisy BVP signal.
4.3 Determine amplitude, frequency and phase

We use this reference signal to determine $A$, $f$ and $\phi$ using a Fast Fourier Transformation with 1024 bins. This results with a frame-rate of 20 fps in a precision of 0.020 Hz or 1.17 BPM per bin. The problem is that our data is only 32 samples so our underlying precision is 38 BPM per bin. In the output of our FFT we get an interpolated version of the coarse precision. This leads sometimes to problems as the highest peak in the spectral domain does not always correspond to the heart-rate. To help the system detecting the correct frequency we use some heuristics.

4.3.1 Frequency adjustment heuristics

As the heart-rate frequency is not changing extremely rapid over time\(^1\) we can use the previous detected frequency as a guideline for detecting the new frequency. This is done in two ways. First the peak detector is limited to the range $(f_{hr,old} - \alpha) ... (f_{hr,old} + \alpha)$. This way the new detected frequency differs at most $\alpha$ Hz from the previous one. The other method slows down even further the heart-rate change as it calculates the new

\(^1\)The heart-rate can in fact change quite rapidly, but under the conditions this system is designed for we do not expect rapid changing heart-rates.
heart-rate as follows

\[
    f_{hr_{new}} = \beta \cdot f_{hr_{detected}} + (1 - \beta) \cdot f_{hr_{old}}
\]  

(4.2)

This limits the heart-rate so that it differs at most \( \beta \cdot \text{abs}(f_{hr_{detected}} - f_{hr_{old}}) \) Hz from the previous one. If we combine these two heuristics we see that the maximal change in heart-rate can be \( \beta \cdot \alpha \). It will be clear that higher parameters \( \alpha \) and \( \beta \) lead to quicker adjustments of the heart-rate but can also cause problems by detecting the correct frequency. Using experiments on a small number of recordings we manually determined the best parameters for non rapid changing heart-rates. We use \( \alpha = 0.5 \) and \( \beta = 0.1 \), which leads to a maximal change of 0.05 per iteration. As this procedure is run every 16 frames and with a frame-rate of 20 fps this leads to a maximal possible change in heart-rate of 0.0625 s\(^{-2}\) or 3.75 BPM/s. This manual optimization process is only performed on a small number of recordings. For future work these parameters should be optimized with a larger dataset and for example a cross-validation mechanism.

### 4.3.2 Increased window

The other option to increase the precision of the heart-rate detection mechanism is to use a bigger dataset in the FFT. We do this by using the reference signals from previous runs and combine these with the trimmed-mean reference of the current run. This leads to a bigger dataset, so our precision of the FFT increases. We use 4 history runs, so instead of 32 frames we have 160 frames of data. Why we use 4 runs of history is a trade-off between precision and data-validity. As we take more history runs the FFT precision would increase, but also the chance of having multiple peaks in the histogram because of a changing heart-rate. With our settings we have a precision of 7.5 BPM per bin over a timeframe of 8 seconds. Also here a shorter history leads to less precision, but capable of tracking faster changing heart-rates.

### 4.4 Synthetic reference

With only the amplitude, frequency and phase of the reference signal we cannot generate a reference signal. We also need a definition of the periodic function, or a defined pulse shape. The pulse shape is defined as one period with amplitude of 1 and a phase of 0. It is saved in memory as a sampled signal, which in our case contains 250 samples. Linear interpolation is used for generating arbitrary frequencies and phases. Now we can generate a synthetic reference for 32 samples which match our previous reference and call this the new reference or synthetic reference.
4.4.1 Pulse shape

We tested several static, pre-defined, pulse shapes. We can divide the tested pulse shapes in three categories. The first category is a triangular shaped pulse based on two line segments, where the steepness of the rising and falling edge is adjustable (Equation 4.3). The second category is similar to the first but replacing the line segments with sinusoids (Equation 4.4). This leads to a more natural shape with a continuous first derivative. An example of both categories is shown in Figure 4.4. The last category is not based on any mathematical formula, but the result of the temporal processing (see Chapter 3) of a recording from the same person. The problem with this last method is that the pulse shape can be specific for a single person, or even a specific body part. That is why we developed a technique that uses a feedback mechanism to update the pulse shape based on the current recording. This feedback mechanism is further explained in Chapter 5. An experiment to qualify the categories of pulse shapes is performed and further discussed in Section 6.1.

\[
\delta(x) = \begin{cases} 
-\frac{2(x-1)}{\alpha-1} + 1 & \text{if } 1 \leq x \leq \alpha \\
\frac{2(x-\alpha)}{n-\alpha} - 1 & \text{if } \alpha < x \leq n 
\end{cases} \quad (4.3)
\]

\[
\delta(x) = \begin{cases} 
\cos\left(\frac{\pi(x-1)}{\alpha-1}\right) & \text{if } 1 \leq x \leq \alpha \\
\cos\left(\frac{\pi(x+n-2\alpha)}{n-\alpha}\right) & \text{if } \alpha < x \leq n 
\end{cases} \quad (4.4)
\]

4.5 Correlate and select signals

Now we have a clean synthetic reference trace we can compute the correlation coefficients of each individual normalized trace with the synthetic reference trace. While calculating the correlation coefficients we allow a maximal lag of ±5 samples to allow a small phase differentiation of the signals between the blocks. We use an unbiased [23] version of the
correlation coefficients and a normalization of the correlation coefficients such that with auto correlation and zero lag the correlation coefficient would be 1. Equation 4.5 shows how the unbiased correlation coefficients are calculated with $m = \{-5, -4, ..., 4, 5\}$ and $N = 32$. Equation 4.6 shows the normalization of the coefficients.

$$c(i,m) = \begin{cases} \frac{1}{N-m} \sum_{n=1}^{N-m} S_i(n + m) \times S_{ref}(n) & \text{if } m \geq 0 \\ \frac{1}{N+m} \sum_{n=1}^{N+m} S_i(n) \times S_{ref}(n - m) & \text{if } m < 0 \end{cases} \quad (4.5)$$

$$c_N(i,m) = \frac{c(i,m)}{c(ref,0)} \quad (4.6)$$

Now we calculated all the correlation coefficients we select those signals which correlate best with the synthetic reference signal. We do this by setting a threshold value at 1.0 for the correlation coefficient and lower this threshold with 0.05 each time until we have selected at least 10% of the traces. This way we make sure that we select all traces with a similar correlation coefficient.

The top row in Figure 4.3 shows the best correlating traces with correlation coefficients of $\{0.94, 0.91, 0.90, 0.89\}$. Where the bottom row shows the traces with the worst correlation coefficients namely: 0.00, 0.01, 0.03, 0.04.

As we allowed some phase-shift we need to correct the phase so all selected signals have the same phase. The problem is that we cannot use a circular shift as we do not have an integer multiple of periods in our traces. We solved this by filling the gap with zeros. This will decrease the absolute value of the first (or last) couple of samples in the mean of the selected signals but this does not influence the output significant. This is because we use an overlap-add method with a Hanning window to combine multiple
traces over time, see Section 2.3.3. By using this Hanning window the first and last
couple of samples per trace have a low weight, so the effect would be minimal.
Chapter 5

Pulse waveform feedback

In the spatial process we make use of a pulse waveform to select traces that correspond with this waveform. As described in Section 4.4.1 we tested several different waveforms. The experiments explained in Section 6.1 it showed that using a pulse waveform based on the temporal processing of the same (or similar) recording leads to good results. It also showed that adaption of the pulse shape to the recording does not degrade the signal quality. This is why we developed a system with a pulse waveform feedback. We can adjust the block diagram in Figure 1.5 and add the feedback, as done in Figure 5.1. This feedback mechanism is implemented in such a way that the processing (spatial and temporal) is done repeatedly for short intervals. This way we can use the pulse waveform we extracted from the current and previous intervals to improve the spatial processing of the next interval.

![Block diagram](image)

**Figure 5.1:** A block diagram describing the building blocks of the complete process including the pulse waveform feedback.
While discussing previous chapters we used some assumptions which we do not need anymore or can relax by doing the processing in short intervals. The first assumption was that the heart-rate is constant over the complete length of the recording. We detected the heart-rate based on the spectral domain of the complete recording. We did allow some variance in the pulse length however this was limited by the detected dominant heart-rate. We do not need this assumption anymore while doing the processing in short intervals because we can recalculate the heart-rate every interval. A second assumption is that the pulse waveform is constant over time. We use this assumption to do the temporal processing on the complete recording and use all detected pulse intervals to calculate an average pulse waveform. While doing the processing in short intervals we can relax this assumption as we can use some kind of sliding window principal to select only the most recent detected pulse intervals. Due to these assumptions we were only able to process stable and relative short recordings where the assumptions were valid. By relaxing these assumptions we do not have a maximal time limit anymore and we can handle varying heart-rates and pulse waveforms.

Besides the benefits as an accurate pulse waveform in the spatial processing and less tight assumptions there are also extra difficulties introduced by using short intervals. These difficulties and our solutions are explained in the following sections.

5.1 Length of a single run

The length of a single run has affect on several parts in the algorithm. The first things to take care of are the artifacts that occur at the beginning and end of a run. Several steps in the processing, including the normalization and peak detection, do not work correct in the beginning or at the end of a run. For the normalization we have the problem that the splines have no start and end point. This way the data is sometimes normalized in a strange way, wasting useful information and making further processing of this part of the data useless. The other part of the algorithm that is affected by the boundaries of a run is the interval or pulse detection. As this detection mechanism is only capable of detecting pulses within the boundaries, pulses that cross the boundaries are not detectable. To solve these problems we use a parameter for each of the processing steps to specify the number of runs to use in each step. So if we specify step \( X \) to use \( m_x \) runs and the current run is \( n \), it will process run \( n - (m_x - 1) \ldots n \) as one set of data. This way we reprocess \( m_x - 1 \) runs of this step, and replace the output of these runs. This way we minimize the effect of the boundaries problems.

As we have a \( m_x \) for every step \( X \) we can make the length of a single run arbitrary short. The disadvantage of this would be that we have to reprocess a lot of data. Making the
length of a single run relative high also causes a lot of reprocess as in most steps we need at least two runs to tackle the boundaries problems. That is why we set the length of a single run to 48 samples, which is the first multiple of 16 greater than 40 (which is 2 seconds at 20 fps). We use a multiple of 16 as this makes sure that the number of steps in the spatial processing, which is done every 16 frames, is equal for every run. We use a period of 2 seconds to make sure that even with the lowest heart-rate of 30 BPM the intervals remain detectable by using $m_{\text{interval}} = 2$.

## 5.2 Introduced difficulties

### 5.2.1 Normalization

With the normalization step where we use splines through the peaks and troughs the starting and ending points of these splines are unknown. The problem with these splines is that at the start and end of a run we do not know the value and derivative of the splines as we do not know the previous or next peak (or trough). We use $m_{\text{normalization}} = 2$ so we can reprocess the possible incorrect ending of the previous run. To solve the problem at the start we use the information from previous run to specify the value and derivative of the beginning of both splines. This is applicable for all but the first two runs, so we will just ignore the start of our dataset.

### 5.2.2 FFT

As explained before, calculating a spectral domain of a short dataset does not give a good accuracy. That is why we use $m_{\text{fft}} = 4$ so we can determine the heart-rate over a period of almost 10 seconds. This gives enough accuracy to determine the frequency while still allowing changing heart-rates.

### 5.2.3 Peak detection

As the normalization can make the data useless at the end, we have to reprocess the last run and $m_{\text{peak}} = 2$. The problem that we introduce by reprocessing the previous run is that we have to replace previously detected peaks if and only if we detected a new peak close to the old peak. We implemented this in such a way that a peak is only replaced by a new detected peak if the distance between those two peaks is smaller than $0.6 \times \frac{10f_{hr}}{f_{fs}}$. If the detected peak does not replace a previously detected peak, it is considered as a new peak.
5.2.4 Interval detection

As a result of the errors in the normalization and peak detection we have to reconsider
the previously detected intervals so we set $m_{\text{interval}} = 2$. Another reason to use more
than one run is the boundary problem where intervals can only be detected within the
boundaries of this part of the data. Also here we have to deal with previously detected
intervals and determine whether to replace them or not. We use the average of the start
and end point of an interval as reference. An old interval is replaced by a new detected
one if the difference between the two reference points is less than $0.3 \times \frac{10 f_s}{f_{hr}}$.

5.2.5 Pulse waveform

The step to normalize and calculate an average pulse waveform is not affected by the
number of runs. Here we use always the 30 most recent pulses to use them in a trimmed-
mean.
Chapter 6

Algorithm validation

In this chapter we show several experiments that we did to validate our proposed algorithm. This includes validations of both the temporal and spatial processing as well as the feedback mechanism. For each of the experiments we explain what we want to test and what we expect. Also, we explain how the experiment is executed and show the results.

6.1 Pulse waveform

For the spatial processing we correlate the individual traces with a reference. This reference is generated based on amplitude, frequency, phase and a pulse waveform. The pulse waveform is a pre-defined shape (in case of no feedback), where all the other parameters are extracted from the current dataset.

6.1.1 Pre-defined shape

As described in Chapter 4.4.1 we initially came up with two different categories, one based on two linear line segments and one based on two cosine segments. In each category we have the parameter $\alpha$ which defines the steepness of the rising and falling edge. We want to test which of the categories gives the best performance in terms of SNR, correlation and consistency between the pulses.

As we look at the raw data we see that the falling edge is the steepest one, but it is only a small difference. So we expect to see a maximum performance per group with an $\alpha$ slightly lower than $\frac{n}{2}$ which in our case for $n = 250$ is somewhere below 125. We also
expect that on average the cosine shaped waveforms would perform better because the BVP signal looks more like a smooth cosine than a piecewise linear function.

The setup for this experiment is to make a recording of one hand of 9 volunteers with a length of one minute. We asked the volunteers to minimize movement. We assume that the pulse waveform and heart-rate remains roughly constant over time in this period of one minute. We do not use the feedback mechanism as we would like to test a pre-defined pulse waveform. For each of the recordings we process the data while varying $\alpha$ in steps of 10 between 10 and 240 for both the linear and cosine shaped functions. We take the average over all the recordings of our performance metrics for each $\alpha$. The results are shown in Figure 6.1. As a reference we also included the SNR and consistency of the two original spatial processing methods, namely a trimmed-mean and an inverse variance weighted.

As we look at the average correlation coefficient between the reference pulse shape and the data we see that the best correlation is achieved with a cosine function with $\alpha = 90$. This corresponds with our expectations. Also the SNR and consistency between pulses performs well with these settings. We can also see that a small mismatch in shape does not influence the SNR and consistency significant. Another important observation is that using our proposed spatial processing performs better than the two original methods in both the SNR and consistency, see also Section 6.3.
Chapter 6. Algorithm validation

6.1.2 Iterative feedback

Instead of using a pre-defined shape as a reference we could also use the pulse waveform extracted from the same dataset using the temporal processing. We will do this in an iterative way, where we first start with a pre-defined pulse shape based on a cosine and $\alpha = 90$. After extracting the pulse waveform from this run we use this extracted waveform as reference input for the next iteration. Again we look at the same metrics as in the previous experiment.

We expect to see improved results over a pre-defined shape as the correlation between reference and data should be better. We use the same data from the 9 volunteers as before and iterate 5 times. Since the results from all volunteers are similar we show the results of one person in Figure 6.2.

From this experiment we can conclude that there is not a significant difference between using a pre-defined shape and a previously extracted pulse waveform. The correlation does increase, but not significantly. The fact that the other metrics do not improve is related to the fact that these metrics can handle a small variation in the pulse shape, due to how the spatial processing is implemented. One could think that using a pulse waveform feedback would not be useful, however this experiment proves that using a previously extracted pulse waveform allows adaption to the correct waveform for a particular recording without a significant decrease in performance.

---

**Figure 6.2**: The results of a single recording of one hand with an iterative pulse waveform feedback.
6.1.3 Complete feedback

This experiment is performed to test if and how fast the complete processing including feedback gives a similar pulse waveform as when using the iterative approach like in the previous experiment. Again we use the same recordings of the 9 volunteers. Instead of iterating the process with each time an updated pulse shape we now adapt the pulse waveform as soon as a new pulse waveform is available from the temporal processing as described in Chapter 5. We expect that at the end of the recording the pulse waveform is converged to the one from the iterative processing. Since again the results are similar for all volunteers we only show one.

Figure 6.3 shows the extracted waveform in this experiment of a single recording. To distinguish the iterative feedback waveform (reference), the last complete feedback waveform and the intermediate waveforms we draw the reference and the last waveform in with a thicker line. As expected the reference waveform and the waveform extracted after 60 seconds of processing are almost similar. The small difference is partly caused by the problems introduced in the feedback processing, see Chapter 5. Another cause is that the iterative processing takes all detected periods into account, where the feedback processing only takes the 30 most recent periods into account. This volunteer had a heart-rate of 47 BPM during the recording, so during the recording of one minute the iterative processing had about 50% more periods available to average. This is also an explanation why the waveform after 48 seconds already gives similar results as the one
Figure 6.4: The pulse waveform extracted from the hand of each volunteer.

after 60 seconds of processing. After 48 seconds 37 periods can be detected where only the 30 most recent are used.

6.1.4 Volunteer related pulse waveform

Allen and Murray [24] have shown that the pulse waveform extracted with a contact PPG sensor differs over individuals. In this experiment we will extract the rPPG pulse waveform with our proposed method from our 9 volunteers. We will make a recording of the hand of each volunteer and use our complete processing including the pulse waveform feedback. During the recordings the volunteers are asked to sit still and minimize any movement. We use the extracted pulse waveform after processing 1 minute of data. Because the contact PPG pulse waveform shows differences between individuals we expect to see these differences as well with rPPG.

Figure 6.4 shows the extracted pulse waveform from our volunteers. It is clearly visible that the rough pulse waveform is similar over all volunteers. But this figure also shows us that all pulse waveforms are unique in some way. Because the origin of the differences in waveform is not clear we also included the heart-beat of each volunteer during the experiment. But even with this information we could not find a clear relation between the pulse waveform and the heart-rate.
6.2 Robustness to noise and interfering signals

In previous experiments we did not compare our method with a golden standard method. This is because we, as far as we know, are the first who try to extract the pulse waveform from remote PPG. That is why we also do not know how the pulse waveform would look like, and whether our proposed method can accurate extract this waveform or is introducing some artifacts. That is why we also performed several experiments with generated data. With this generated data we know how the waveform should look like. This way we can compare our extracted waveform with the input and by calculating the Euclidean distance between the two waveforms we have a metric we can use. With the metric that gives us information about the extracted waveform we can also test to what extend our system is capable of extracting a correct waveform on input signals containing noise or interfering signals.

Because we want to test the complete system we need to generate the color traces that are used as input for our processing, see also Chapter 2.3. According to de Haan and Jeanne [17] the ratio between the R, B and G channel of the BVP signal is \( r = \{0.3, 0.8, 0.5\} \). Since the DC component of the color signals is much bigger than the BPV component we constructed the color channels as \( C(t) = 100 + r_C \times BVP(t) \) where \( C \in \{R, G, B\} \). Using this method we can generate color signals that contain the BVP signal similar to real camera recorded signals. For the BVP signal we can use several types of functions. The two functions tested in this experiment are a sinus wave and a cosine based pulse shape with \( \alpha = 90 \) (see Chapter 4.4.1). Both functions are generated with a frequency and amplitude of 1. Because we also want to test our proposed spatial processing we need multiple traces so we can actually do some spatial processing. That is why we generated 100 similar traces, which is a rough average of the recordings seen so far.

6.2.1 Noise

To test whether our system is robust to noise we generated a uniform noise set which we will use in every test. The noise is generated as follows:

\begin{algorithm}
\caption{Noise generation}
\begin{algorithmic}
\For {\( t \leftarrow 1, \text{NumberOfTraces} \)}
\State \( \text{TraceLevel} \leftarrow \text{rand}(1) \)
\For {\( n \leftarrow 1, \text{NumberOfSamples} \)}
\For {\( c \leftarrow 1, 3 \)}
\State \( \text{Noise}_{t, n, c} \leftarrow \text{TraceLevel} \times \text{rand}(1) \)
\EndFor
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}
where \( \text{rand}(1) \) will generate a random number in a uniform distribution between 0 and 1. By using this algorithm for noise generation each trace has its own level of noise, which is also the case in real recordings. The generated noise has a maximal amplitude of 0.5, which is half of the signal.

In this experiment we use the generated signal and add different levels of noise to the signal and try to extract the pulse waveform. We compare this extracted waveform with the waveform we used for generating the signal and calculate the Euclidean distance between these two. We execute the experiment for two spatial processing methods, the original spatial processing (inverse variance weighted) and our new proposed spatial processing. We will also take a look at the SNR of the BVP signal of both methods. We run the experiment for 8 times per method with the first time no additive noise, the second run we add the generated noise once. Every next run we double the generated noise, up till the point where the noise is at level 64, which means that the amplitude of the noise is 32 times higher than the signal.

Figure 6.5 shows the results of this experiment while using a sinus waveform as signal. The other tested waveform shows similar results. One of the observations is that the Euclidean distance is roughly constant until the noise level becomes higher than 8. At this point both methods start to produce waveforms that do not look like the input signal anymore (from visual inspection). In the lower left plot the waveforms are shown with a noise level of 8, which shows that both methods produce a similar waveform as the input, again with visual inspection. This shows that our temporal processing is capable of extracting a clean pulse waveform even with noise with an amplitude of 4 times the signal.

### 6.2.2 Noise and interfering frequencies

If we look at our previous experiment there is not a good explanation to use our proposed spatial processing. It does produce a higher SNR but this does not lead to more robustness for noise if we look at the extracted pulse waveform. Actually in real recordings we do not see only random noise, but also interfering signals appear in the recordings. These interfering signals can come from a variety of sources. An example source of an interfering signal is movement. If we take a recording from a person doing an exercise on a treadmill we would see the person’s head moving up and down on a constant frequency caused by the steps of the person. The blocks at the edge of the person’s face would contain the BVP signal as well as the frequency caused by the movement. As our proposed spatial processing is capable of selecting the traces based on a frequency
(and other parameters), it probably is capable of selecting those traces that does only contain a small level of the movement frequency.

For this experiment we generated an interfering signal using a sinus function at a frequency of 1.7 Hz and an amplitude of 0.5. We select 1.7 Hz as it is not an integer multiple of the BVP frequency, which would look like as a harmonic of the BVP signal. Also 1.7 Hz is within the range of the possible heart-rate, so this frequency cannot be filtered out using a bandpass filter. The interfering signal is put in the color signals similar to the BVP signal, so with the ratios $r = \{0.3, 0.8, 0.5\}$. For 75% of the traces a uniform random level is selected between 0 and 1. For the other 25% no interfering signal is generated.

In this experiment we use the same scheme as with the experiment with only noise. The only difference is that we add both an interfering signal and noise. We use the same noise as in previous experiment, but divided by two because the interfering signal should be dominant.

Figure 6.6 shows the result for the experiment with an interfering signal and noise. In contrast to the previous experiment we now see a clear difference between the original spatial processing and our new proposed method. Where with our method the SNR remains roughly constant (difference of at most -0.5 dB) at the different levels of interfering signals and noise, the original method quickly suffers from the added noise and interfering signal (difference of at most -7.0 dB). This is also clearly visible if we look at
Chapter 6. **Algorithm validation**

6.3 SNR improvements

In the experiments with a generated signal we show that our proposed method is more robust to interfering signals compared to the original methods in terms of SNR and consistency of the pulse waveform. An SNR difference of 6.4 dB and a difference in the Euclidean distance of 1.00 between the original and proposed method is shown in

The metrics after processing a synthetic signal with different levels of additive noise and interference. Waveforms are shown at noise level 4.
favor of our proposed method. We also calculated the SNR for the 9 recording from our volunteers. This shows an average increase of $\mu = 1.55$ dB with a standard deviation of $\sigma = 1.55$ dB compared to the original methods. Figure 6.7 shows the SNR of each of the methods of all volunteers. In 8 out of 9 cases a higher SNR is achieved with our new proposed method, in 1 case the trimmed-mean method performs best. Table 6.1 gives the results of this experiment.

To test whether this improvement is significant we used a paired student’s t-test. This

![Figure 6.7: The SNR results of the different spatial processing methods on all volunteers.](image)

**Table 6.1:** The SNR results of the different spatial processing methods. Method I refers to the inverse variance weighted method, II refers to the trimmed-mean method and III refers to our new proposed method based on correlation. All values are in decibel (dB).
test shows whether the two methods are really different or that the observed differences can be caused by random variation. According to the 95% confidence interval of the difference, our proposed method shows an improvement between $[0.71, 2.38]$ compared to the inverse variance weighted method with a $p = 0.003$ and an improvement between $[0.36, 2.75]$ compared to the trimmed-mean method with a $p = 0.017$. So based on these experiments we can state that our proposed method, with a 95% confidence, improves the signal quality in terms of SNR over the other methods. We also looked at the difference between method I and II and performed the paired student’s t-test on these two methods. As $p = 0.975$ for this test we can say that these two methods perform similar in terms of SNR.
Whereas photoplethysmography (PPG) is used for many years in the medical area, remote PPG (rPPG) is relatively new. Recently, the research on rPPG increased in popularity, where most research is about improving motion robustness. The pulse waveform of a single period of the PPG signal is used in various medical applications. We think that it would be meaningful to be able to extract the pulse waveform with a camera. That is why we proposed a method to extract the pulse waveform from a rPPG signal. To the best of our knowledge we are the first who try to extract this pulse waveform from rPPG and so we have no golden standard reference to compare our extracted waveforms with. We tested our method with generated data and its robustness to noise and interfering signals. These experiments show that our method is capable of extracting the pulse waveform under various conditions. Next to extracting the pulse waveform we also used this extracted waveform to improve the rPPG signal. We proposed a new method for the spatial processing of the rPPG signal where the pulse waveform is used as a selection criterion. This is done with a feedback mechanism where the pulse waveform is updated during the processing. This new proposed spatial processing showed good results on the generated data with interfering signals and noise. When we run the new proposed method on recordings of the hands of 9 volunteers we see an improvement in 8 out of 9 cases in terms of SNR. The average improvement in terms of SNR is $\mu = 1.55/1.55$ dB compared to inverse variance weighted / trimmed-mean method. The spread of the improvement is $\sigma = 1.09/1.55$ dB and the results of a paired student’s t-test show a statistical significant improvement with $p = 0.003/0.017$. 

$\mu = 1.55/1.55$ dB compared to inverse variance weighted / trimmed-mean method. The spread of the improvement is $\sigma = 1.09/1.55$ dB and the results of a paired student’s t-test show a statistical significant improvement with $p = 0.003/0.017$. 

43
7.1 Future work

In the experiments for comparing the SNR we used a peak detector in the spectral domain to determine the heart-rate and used this peak as signal in the SNR calculation. A good approach would be to simultaneously use an ECG and rPPG signal. Whereby you could use the detected heart-rate from the ECG to calculate the SNR of the rPPG signal. Also several parameters in both the temporal and spatial processing are chosen arbitrary (nevertheless performing well) and can probably be optimized by using some sort of cross validation. Furthermore extra research should be performed on the actual meaning and possible use of the extracted pulse waveform from rPPG. A good experiment would be to test whether the features that are contained in the pulse waveform from PPG are also available in the pulse waveform of a rPPG signal.
Bibliography


