Developing a Non-invasive, Camera-based Method of Measuring Blood Pressure

Samuel Tober
samueltober@gmail.com

under the direction of
Prof. Hamed Hamid Muhammed
Departement of Technology and Health
Royal Institute of Technology

Research Academy for Young Scientists
July 13, 2016

Abstract

Blood pressure (BP) is one of the most important physiological parameters and a vital indicator of a person's health, thus making it hugely important to monitor. This paper presents a new method of estimating BP using a consumer grade camera and image processing in MATLAB. Four test subject's arms were filmed following different physical activities. Owing to the fact that the BP is linearly related to the pulse wave velocity (PWV) the BP can be estimated through analysis and processing of the films. Results indicate that the method works albeit, with a relatively low accuracy having an average error of 16.95 $\pm$ 1.84 % for the systolic BP and 10.75 $\pm$ 3.0 % for the diastolic BP, compared to a conventional cuff monitor. Although able to produce moderatly accurate results, the method is in need of further improvements.

Contents

1 Introduction

Blood pressure (BP) is one of the vital indicators regarding personal healthcare and it has been proven useful in detecting several dieases and defects [1]. BP is defined as the pressure which the blood exerts upon the walls of the blood vessels [2]. Most commonly, BP is expressed as the systolic pressure (the pressure that arises from contraction of the heart) over the diastolic pressure, (the pressure that arises from the expansion of the heart) measured in mmHg [3].

A high BP can provoke arterial strains and over time the arterial tissue can be damaged, whereas a low BP results in fatiuge and weariness. It is therefore crucial to monitor BP regularly [4]. There are many ways of measuring BP however, most methods are not optimal regarding patient comfort as well as accuracy.

Many attempts of measuring BP and heart rate using various optical sensors and cameras have been made. In a paper published by Wu et al. a method used to amplify subtle colour changes was presented. With this method they were able to detect colour changes caused by the flow of blood through vessels in the head, thus being able to see the pulse propagate through parts of the body. [5]

Anohter group of scientists, M. Younessi and M. Khalilzadeh, tested a BP measuring method using optical sensors. Their approach was to measure the PTT using infrared sensors and then measure the BP after various physical activities. Coefficients showing the correlation between PTT and BP were calculated and the models were tested. [1]

In this study a new method of measuring BP is presented and evaluated. By combining the two methods described above, using a camera instead of the optical sensors used by M. Younessi and M. Khalilzadeh, a more practical BP measuring method can be developed. Such a non-contact method would improve both comfort and availibility for the patients, being a step towards an improved health care.

2 Theoretical Overview

2.1 Methods of Measuring Blood Pressure

Today there are numerous ways of measuring blood pressure. Generaly, these are categorised into invasive and non-invaise methods.

When conducting an invasive BP measurement, a needle is inserted directly into an artery. This arterial line is connected to electronic tranducers through which the blood pressure can be monitored continously. Even though the invasive methods have a comparatively high accuracy, these methods can be burdensome for the patients since they involve the insertion of an arterial line into the arm. Moreover, invasive methods can cause bleeding as well as an increased infection risk. [6] Non-invasive methods are in general more simple, more comfortable for the patients and pose lower risks of complications. On the other hand, they are generaly less precise. Although they have a relatively low accuracy they do yield results that are accurate enough for routine examinations. This is why the non-invasive methods are the most frequently used. [1]

An example of a non-invasive method is the sphygomanometer, more commonly know as the cuff method. Cuff-based methods are relatively simple and comfortable, however, the method is not ideal since it cannot measure the blood pressure continously. Furthermore, a sphygmomanometer has to be serviced and recalibrated often in order to provide accurate results. [7]

2.2 Pulse Transit Time and Pulse Wave Velocity

There is a strong dependency between the BP and the Pulse Transit Time (PTT) [8]. The PTT is the time required for a pressure wave to travel from the heart to another place in the body within the same cardiac cycle (from the beginning of one heartbeat to the start of the following). The difference in PTT, denoted by PTTD, can be measured directly by comparing two points (such as a toe and a finger) [9]. The process of measuring PTTD will be explained more thoroughly later in this paper.

The Pulse Wave Velocity (PWV) is defined as the rate at which the blood pressure moves down a blood vessel [9]. The PWV is related to the stiffness of arteries and has a direct connection to the BP; it has been shown that an increase in PWV implies an increase in BP [10]. The relation between PWV and BP can be described:

$$PWV = \sqrt{\frac{V\Delta BP}{\rho \Delta V}} = \frac{1}{cBP - c/4} \tag{1}$$

where ΔBP and ΔV represent change in BP and blood volume, ρ is the density of the blood and c is a normalising constant. Furthermore, the relation between PWV and PTTD is:

$$PWV = \frac{d}{PTTD} \tag{2}$$

where d is the distance between two points (for example from a finger to a toe). By combining Equations (1) and (2) and using a Taylor series expansion, a linear relation between BP and PTTD is obtained:

$$PTTD = d(cBP - c/4) \Rightarrow BP = A + (PTTD)B \tag{3}$$

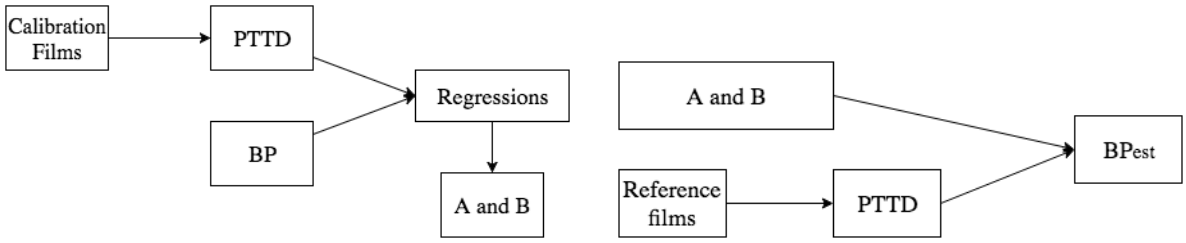
where A and B are constants that depend on the specific individual and therefore must be determined for each test subject. [1]

2.3 Colour Images

Colour images are two dimensional matrices consisting of vectors called pixels, containing three values; red, green and blue. A colour image can thus be separated into three different layers: one red (R), one green (G) and one blue (B). Combinations of these three colour layers can yield all the colours in the visible spectrum and are the building blocks of all colour images [11]. Each of these components can have a value between 0 and 1. Red is for example represented by the triplet $\{1, 0, 0\}$ and white is represented by $\{1, 1, 1\}$ [12].

3 Method

The method used to estimate BP in this study can be summarised with the two flowcharts shown below in Figure 1:



(a) The constants A and B are determined using calibration films and BP values measured with a cuff-based BP monitor.

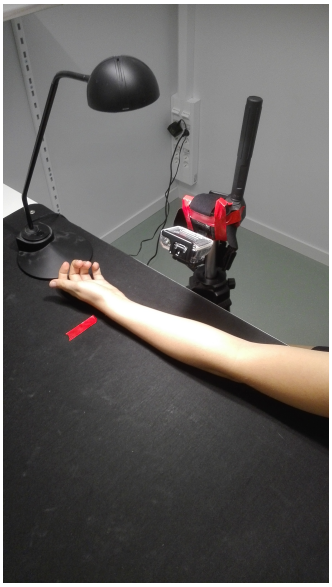
(b) The constants A and B are used along with the PTTDs obtained from reference films to estimate the BP, denoted BP_{est} in the figure.

Figure 1: Flowchart diagram, overlooking the method used to estimate BP.

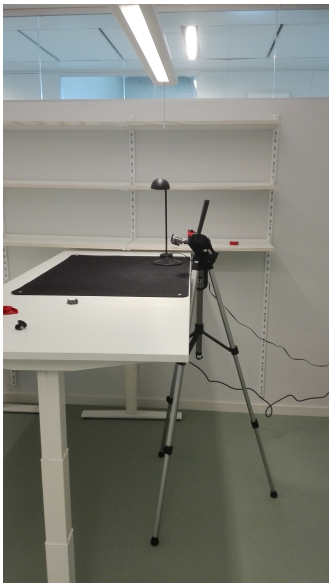
3.1 Data Collection

BP data and films were collected from two males and two females, all subjects being around 18 years old. In order to vary the blood pressure all measurements were conducted after various physical activites, such as running up and down stairs and doing push-ups. All video data was collected indoors with ambient lighting and a desktop lamp aimed at the arm, as shown in Figure 2a. The videos were shot with a GoPro Hero 3 Silver Edition camera set to 60 frames per second (fps) at 720p resolution. All videos were filmed on a desktop with a black background, approximately 15 cm above the table's surface, the duration of each video was roughly 30 s. Upon processing, all videos were separated into individual frames using Adapter Video Converter [13].

In order to determine the coefficients A and B presented in Equation 3 one must obtain BP data and corresponding PTTDs. After capturing each film, the BPs of the subjects were therefore measured. All BP measurements were conducted using a cuff-based BP monitor (Riester ri-champion N Digital). Five measurments were done for each subject, filming the arm as shown in Figure 2a.



(a) Setup used when obtaining all films, close up.



(b) Setup used when obtaining all films, from a distance.

Figure 2: Experimental Setup, showing how the filming of the arm was done.

3.2 Data Processing

To calculate the PTTDs two points of interest on the arm were chosen for comparison, as shown in Figure 3. The two points of interest were cropped to 100 × 100 pixels each. The frames from both points were processed in MATLAB; R, G and B values of each pixel were extracted from each frame and averaged.

Since there is some distance between the two points the colour shift caused by the blood flow will reach the two points at different times. First, point 1 will be reached, after the PTTD has passed, point 2 will be reached. By analysing the shift between the values extracted from point 1 and 2, one can determine the PTTD. In order to do this the averages first need to be normalised to have a mean around zero:

$$x_i'(t) = \frac{x_i(t) - \mu_i}{\sigma_i} \tag{4}$$

where $x_i(t)$ is the R, G or B average in frame t, μ_i is the mean of $x_i(t)$, σ_i is its standard deviation and $x_i'(t)$ is the normalised value in frame t. The normalised values were then run through a custom MATLAB algorithm (Appendix A) to find the PTTD.

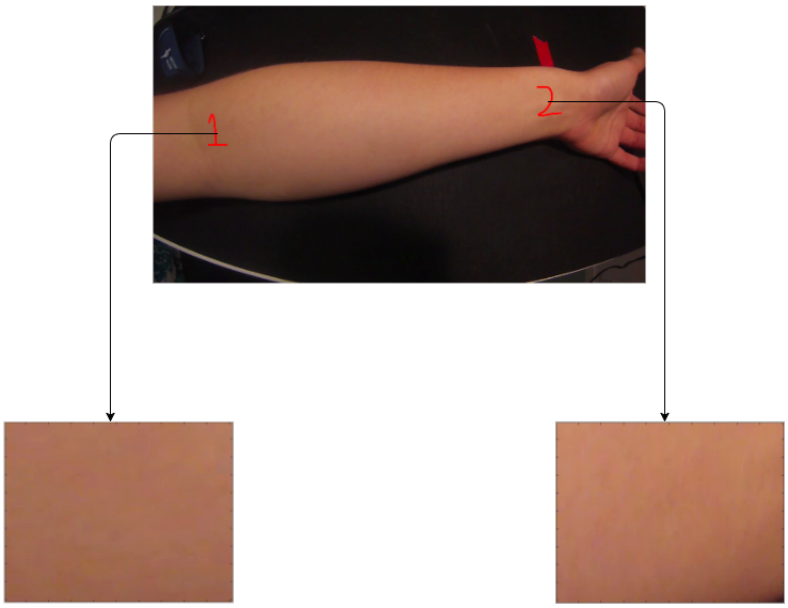
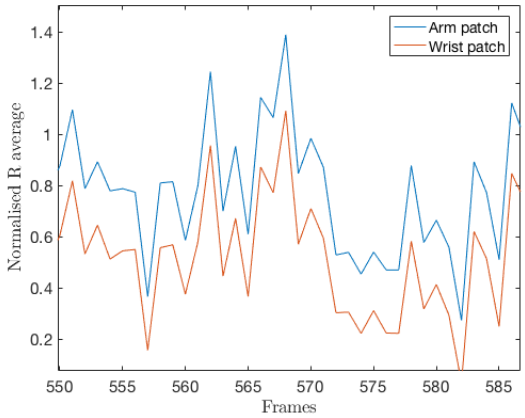


Figure 3: Two points of interest (100 × 100 pixels) are manualy cropped in MATLAB, one from the bend of the arm, and one from the wrist.

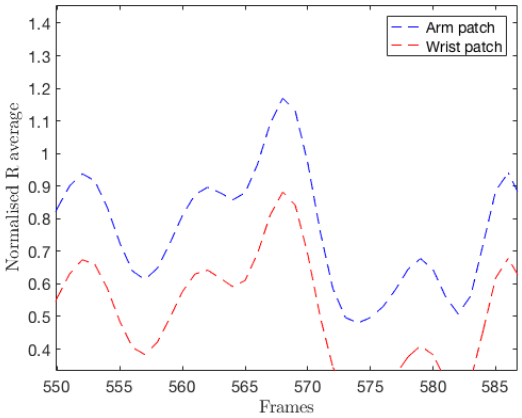
3.3 Analysis Steps

The following is a description of the MATLAB algorithm, converting the normlised values to a PTTD in ms (see Appendix A for explicit details regarding the algorithm).

In Figure 4a, the average of the red (R) colour layer in each frame is plotted against the frame number. The blue curve represents the R values from point 1 in Figure 3 and the red curve the R values from point 2 in the same figure. Given these two curves it is possible to find a horisontal shift in frames between the two, a shift caused by the pulse reaching one point before it reaches the other. However, calculating this shift is difficult given the rough curves presented in Figure 4a. To resolve this problem the data points are approximated using a Fourier series, yielding the smoothened curves shown in Figure 4b.



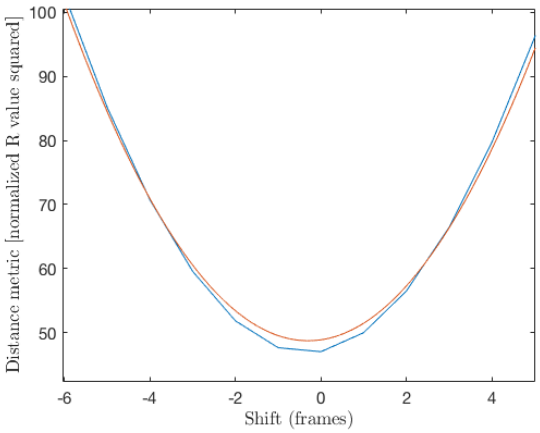
(a) Averaged and normalised values from the red colour layers of the two points presented in Figure 3, plotted for each frame.



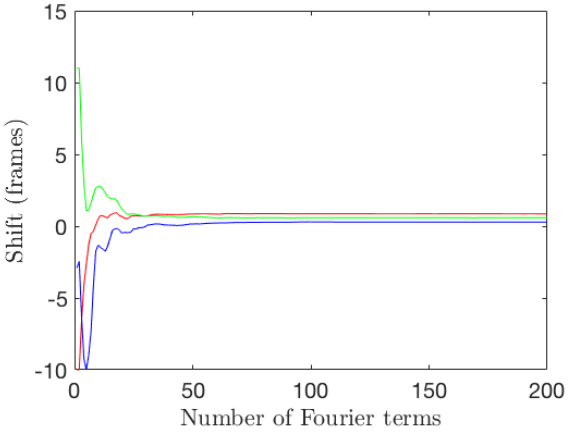
(b) Averaged and normalised values from the red colour layers of the two points presented in Figure 3 approximated using a Fourier series, plotted for each frame.

Figure 4: Averaged and normalised values from the red colour layers of the two points presented in Figure 3 plotted for each frame before and after Fourier series approximation.

To determine the duration of the shift, the two curves are moved horisontally, one frame at a time, and the sum of the squares of the distances between the two curves is plotted after each frame. This yields a curve, to which a second degree polynomial is fitted, see Figure 5a. The latter is done in order to achieve a more precise value compared to the values given by the number indices. The minimum of the parabola is the point were the two curves are as close as possible. The shift required to reach this point is the PTTD in frames. The number of frames shifted at each minimum is plotted, which is shown in Figure 5b. Subsequently this is done for all R, G and B values, generating three curves. Next, the entire process is repeated with more terms in the Fourier series. As is apparent, the curves stabilise close to a certain value for high numbers of Fourier terms; the PTTD. Finally, the PTTD is converted from frames to ms.



(a) A second degree polynomial plotting the sum of the square of the distances between the curves, as a function of how many frames the curves have been shifted horisontally.



(b) Number of frames the R, G and B curves need to be shifted to minimise the distance between them, this value is equal to the PTTD.

Figure 5: Averaged and normalised values from the red colour layers of the two points presented in Figure 3 plotted for each frame before and after Fourier series approximation.

Using MATLAB's polyfit function, the PTTDs obtained from each subject's videos were plotted against their coherent BP values and fitted to the linear model shown in Equation 3. In this way coefficients A and B are derived. Therefrom, in the aspect of accuracy, each model was evaluated using five reference films from each subject. Two models are fitted for each subject; one for the systolic BP (SBP) and one for the diastolic BP (DBP). In Figure 6 an example of a regression between diastolic BP and PTTD is shown.

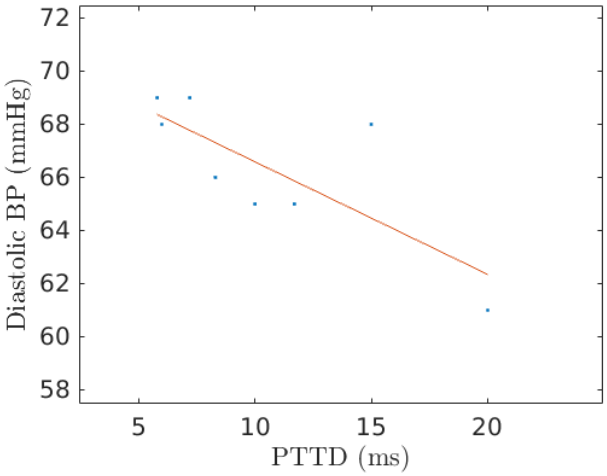


Figure 6: An example of a linear regression, plotting diastolic BP and PTTD taken from one of the test subject's data. In this manner constants A and B are derived.

4 Results

Both the systolic and diastolic correlation coefficients (A_{sys}, B_{sys}, A_{dia} and B_{dia}) were determined for each subject. In Table 1 the correlation coefficients for systolic BP and PTTD are shown for each subject along with error calculations. In the same manner, Table 2 contains the coefficients for the diastolic BP of each subject. Additionally, the mean and standard deviation (σ) was calculated for all data.

Table 1: Constants A and B, showing the correlation between systolic BP (SBP) and PTTD with error calculations. SBP_{cuff} is the systolic BP measured with the monitor and SBP_{est} is the systolic BP estimated with the model.

Subject	A_{sys} [mmHG]	B_{sys} [mmHG/ms]	$\lvert SBP_{cuff} - SBP_{est} \rvert$[mmHG]	Error (%)
1	135.4	-1.1	14.9	16.3
2	125.3	-0.7	22.6	19.7
3	100.3	3.2	18.6	16.0
4	138.4	0.4	19.0	15.8
Mean $\pm$ σ	124.85 $\pm$ 14.98	0.49 $\pm$ 1.68	18.77 $\pm$ 3.15	16.95 $\pm$ 1.84

Table 2: Constants A and B, showing the correlation between diastolic BP (DBP) and PTTD with error calculations. DBP_{cuff} is the disatolic BP measured with the monitor and DBP_{est} is the diastolic BP estimated with the model.

Subject	A_{dia} [mmHG]	B_{dia} [mmHG/ms]	$\lvert DBP_{cuff} - DBP_{est} \rvert$ [mmHG]	Error (%)
1	70.8	-0.4	4.5	7.2
2	70.5	-0.3	14.4	11.1
3	87.2	-0.7	7.9	10.3
4	81.0	-0.3	9.7	14.4
Mean $\pm$ σ	77.38 $\pm$ 8.17	-0.43 $\pm$ 0.17	9.13 $\pm$ 4.12	10.75 $\pm$ 3.0

5 Discussion

5.1 Result Analysis

For the systolic BP the results show an average difference, between the estimate and the cuff measurement, of approximately 19 mmHg which translates to an average error of 17%. The diastolic BP estimates were determined with a higher accuracy, with an average difference of 9 mmHg and a mean error of roughly 11%.

The results show a maximum error of 20 % for SBP and 14 % for DBP which is acceptable when only a moderatly accurate estimation is needed. Although the method might have potential, based on the results, it is not yet precise enough for medical standards.

It is apparent that the coefficients (A and B) vary depending on the test subject. Exactly what causes this variation is not obvious. Many factors, such as age, diet, gender, height and weight, most likely have an influence. On the other hand, these variations might as well be a consequence of errors in the experiments such as insufficient lighting or movement while filming. Another trend that was observed was that the BP seems to decrease as the PTT increases. This is what one would expect judging by the fact that BP increases linearly in respect to the PWV.

5.2 Sources of Error and Flaws

In contrast to other studies this study has one major limitation; that is the cuff method utilised during the calibrations. This is a major source of error since the accuracy of the BP estimation is directly dependent on the accuracy of the calibration. Moreover, when calibrating the system more films should have been made; five subjects is not enough to establish a result with high accuracy.

An additional source of error is that during the 30 s time of filming prior to the BP measurement, the BP will change thus making the measured BP and obtained PTTD slightly incoherent.

The algorithm used to compute the PTTD was found to be computationally intensive and inefficient, taking almost 30 minutes to run through one 30 s film. Moreover, the program could only run through images, meaning that the films first had to be converted into images, taking up unecessary amounts of space.

5.3 Future Work

A question worthy of investigation pertains in the problem issued above; the algorithm utilised to convert a film of the arm to a PTTD value is in need of optimisation. Furthermore, the algortihm involves a few non-automated steps, for one, the cropping of the images as shown in Figure 3 has to be done manually. This is a major limitation since it prohibits an automation of the process.

Another topic of future research is the factors affecting the coefficients. M. Younessi and M. Khalilzadeh have already studied how age affects the coefficients and they concluded that the PTT was linearly related to the age of the subject, decreasing as the age increases [1]. However, there are far more factors yet to be examined and if these factors are accounted for the estimation models can be made even more complex, yielding even more precise estimations.

Something to consider is if it were possible to find general correlation coefficients, i.e. coefficients that were applicable on any person. However, estimating BP with such coefficients could never yield a value as precise as one from an individually adjusted model. Nonetheless, it would facilitate the process to a great extent.

In this study data obtained from the measurements were adapted to a linear model as shown in Equation 3, however this is not necessarily the best adaptation. Some authors have reported a non-linear correlation between the PTT and BP [14, 15]. This is a topic that could be investigated further to optimise and amend the estimations even more.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Hamed Hamid Muhammed for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research. Secondly, I thank my labmate Julia Jansson for being such as wonderfully positive and enthusiastic project partner. It has been a great pleasure working on this project with her. I am also deeply grateful for Europaskolans support during this project. Thank you for allowing us to use your premises. Thirdly, I owe a great thanks to the whole Rays counsler group, Anna Broms, Serhat Aktay, Klara Kiselman and especially Isak Nilsson for all Matlab support. Finally, I would like to thank Astra Zeneca for being a partner of Rays for Excellence.

References

[1] Heravi M, Khalilzadeh M. *Designing and Constructing an Optical System to measure Continuous and Cuffless Blood Pressure Using Two Pulse Signals* [Online]. Iranian Journal of Medical Physics; 2013. [cited 2 July 2016].

[2] Wyngaarden J. B., SmithL. H., *Textbook of Medicine*, 16 edition. Philadelphia: W.B. Saunders Company; 92-94, 105-106, 125-126, 211-220 (1982)

[3] *Blood Pressure Numbers: What They Mean.* NIH MedLine Plus. 2010; (5): 10.

[4] American Heart Association. *About High Blood Pressure*; [revised 2016 Jun 29; cited 2016 July 11]. Collected from: `http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/AboutHighBloodPressure/What-is-High-Blood-Pressure_UCM_301759_Article.jsp#.V4ISDpOLR-U`

[5] Poh, Ming-Zher, Daniel J. McDuff, and Rosalind W. Picard. *Noncontact, Automated Cardiac Pulse Measurements using Video Imaging and Blind Source Separation.* Optics Express 18 (2010): 10762. 2011 Optical Society of America.

[6] Murray IP. *Complications of Invasive Monitoring* [Online]. PubMed; 1981. [cited 2 July 2016]. Collected from: `http://www.ncbi.nlm.nih.gov/pubmed/7231245`

[7] M J Burke et al. *Sphygmomanometers in hospital and family practice: problems and recommendations* [Online]. British Medical Journal; 1982. [cited 2 July 2016]. Collected from: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1499274/pdf/bmjcred00620-0015.pdf`

[8] Maria J. *Continuous non-invasive blood pressure estimation* [dissertation on the Internet]. Zurich: ETH; 2011 [cited 2016 Jul 13]. Available from: `http://e-collection.library.ethz.ch/eserv/eth:5572/eth-5572-02.pdf`

[9] Y A Foo, S J Wilson, G Williams, M-A Harris and D Cooper *Pulse transit time as a derived noninvasive mean to monitor arterial distensibility changes in children.* Journal of Human Hypertension (2005).

[10] TENSIOMED. Pulse Wave Velocity [Online]. Budapest: TENSIOMED [cited 2016 Jul 13]. Available from: `http://www.tensiomed.com/en/pulse-wave-velocity`

[11] Bista S., *Preliminary work on mobile application framework for automated diagnosis of skin cancer* Bachelor's thesis, Stockholm, Royal Institute of Technology (2015)

[12] Jensen J. Hannemose M. *Camera-based Heart Rate Monitoring* [Bachelor of Science in Engineering]. Kongens Lyngby: Technical University of Denmark; 2014 [cited 2016 Jul 13]. Available from: `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/6796/pdf/imm6796.pdf`

[13] Devor C. Adapter, 2.1.6 [computer program]. McKinney, Texas Macroplant LLC; 2008. [2016 Jul 13]. Available from: `https://www.macroplant.com/adapter/`

[14] Chen Y, Wen C, Tao G, Bi M, Li G. *Continuous and noninvasive blood pressure measurement: a novel modeling methodology of the relationship between blood pressure and pulse wave velocity.* [Online]. PubMed; 2009. [cited 11 July 2016].

[15] Dingchang Zheng, Alan Murray *Non-invasive quantification of peripheral arterial volume distensibility and its non-linear relationship with arterial pressure* [Online]. Journal of Biomechanics; 2009. [cited 11 July 2016].

A Matlab Code

The code consists of three parts; one function used to crop the image, extract RGB values and average them, one to load the images, and finally, a main code used to loop through all frames and finding the PTTD. See comments in the code for explicit details.

A.1 Main Code

```
for i=1:1200
    pic=Load(i); %Calls function "Load" to load an image
    rgb=RGB(pic,100,200); %Zooms the image given the x coordinates

    r_1(i)=rgb(1,:); %Stores the r, g and b values in lists
    g_1(i)=rgb(2,:);
    b_1(i)=rgb(3,:);
end
```

```
for i=1:1200 %Same as above except for the wrist patch
    pic=Load(i);
    rgb=RGB(pic,900,1000);

    r_2(i)=rgb(1,:);
    g_2(i)=rgb(2,:);
    b_2(i)=rgb(3,:);
end
```

```
21 R1 = r_1;
22 R2 = r_2;
23
24 R1 = (R1-mean(R1))/std(R1); %Normalises the R values
25 R2 = (R2-mean(R2))/std(R2);
26
27 framediffsR = []; %Creates a matrix for later use
28
29 for i = 1:200
30     [a1,b1] = Fseries(1:length(R1),R1,i);
31     fit1 = Fseriesval(a1,b1,1:length(R1));
32     %Approximates a curve for the data from point 1 using Fourier
       series
33
34     [a2,b2] = Fseries(1:length(R2),R2,i);
35     fit2 = Fseriesval(a2,b2,1:length(R2));
36     %Approximates a curve for the data from point 2 using Fourier
       series
37     figure(1)
38
39     diffs = [];
40     for j=-10:10
41         diffs = [diffs,sum((fit1(20+j:end-20+j)-fit2(20:end-20)).^2)];
42     end
43     %Shifts the curves and saves the sum of the squares of the
       distances between the two curves in variable "diffs"
44
45     parabola = polyfit(1:length(diffs),diffs,2);
46     inp = linspace(1,length(diffs),10000);
47     appr = parabola(1)*inp.^2 + parabola(2)*inp + parabola(3);
```

```
48      %Fits a parabola to the data in "diffs"
49
50      [M,I] = min(appr);
51      I = I/length(inp)*21-10;
52
53      framediffsR = [framediffsR,I];
54      %Finds the minimum distance between the curves and saves it in a
        list
55
56  end
57
58  figure(18); hold off;
59  plot(framediffsR,'r'); hold on;
60  ================================================================================

61  %Same as above except for the green values
62  G1 = g_1;
63  G2 = g_2;
64
65  G1 = (G1-mean(G1))/std(G1);
66  G2 = (G2-mean(G2))/std(G2);
67
68  framediffsG = [];
69
70  for i = 1:200
71      [a1,b1] = Fseries(1:length(G1),G1,i);
72      fit1 = Fseriesval(a1,b1,1:length(G1));
73      [a2,b2] = Fseries(1:length(G2),G2,i);
74      fit2 = Fseriesval(a2,b2,1:length(G2));
75      figure(1)
```

```
76
77      diffs = [];
78      for j=-10:10
79          diffs = [diffs ,sum((fit1(20+j:end-20+j)-fit2(20:end-20)).^2)];
80      end
81
82      parabola = polyfit(1:length(diffs),diffs,2);
83      inp = linspace(1,length(diffs),10000);
84      appr = parabola(1)*inp.^2 + parabola(2)*inp + parabola(3);
85      [M,I] = min(appr);
86      I = I/length(inp)*21-10;
87
88      framediffsG = [framediffsG,I];
89
90  end
91
92  figure(18);
93  plot(framediffsG,'g'); hold on;
94  ================================================================================

95  %Same as above except for the blue values
96  B1 = b_1;
97  B2 = b_2;
98
99  B1 = (B1-mean(B1))/std(B1);
100 B2 = (B2-mean(B2))/std(B2);
101
102 framediffsB = [];
103
104 for i = 1:200
```

```
105     [a1,b1] = Fseries(1:length(B1),B1,i);
106     fit1 = Fseriesval(a1,b1,1:length(B1));
107     [a2,b2] = Fseries(1:length(B2),B2,i);
108     fit2 = Fseriesval(a2,b2,1:length(B2));
109     figure(1)
110
111     diffs = [];
112     for j=-10:10
113         diffs = [diffs,sum((fit1(20+j:end-20+j)-fit2(20:end-20)).^2)];
114     end
115
116     parabola = polyfit(1:length(diffs),diffs,2);
117     inp = linspace(1,length(diffs),10000);
118     appr = parabola(1)*inp.^2 + parabola(2)*inp + parabola(3);
119     [M,I] = min(appr);
120     I = I/length(inp)*21-10;
121
122     framediffsB = [framediffsB,I];
123
124 end
125
126 figure(18);
127 plot(framediffsB,'b'); hold on;
128 ================================================================================
```

A.2 Load Image Function

```
function [rgb] = Load(i)
    t(i,:)= ['a' num2str(i) '.png'];
    filename=['/Directory/ t(i,:)'];
    %Defines the directory in which the images are stored

    y=imread(filename);
    %Reads image

    y=double(y);

    b=y(:,:,3);
    g=y(:,:,2);
    r=y(:,:,1);
    %Seperates R, G and B values

    r=r/max(max(r));
    g=g/max(max(g));
    b=b/max(max(b));
    %Scales the values

    rgb(:,:,1)=rr;
    rgb(:,:,2)=gg;
    rgb(:,:,3)=bb;
    %Reassembles the image

end
```

A.3 RGB Extraction Function

```
function [rgb] = RGB(bild,x_1,x_2)
    im=bild;
    %Reads the image

    y_1 = 400;
    y_2 = 500;
    %In what interval should the image be cropped

    bildzoom=im(y_1:y_2,x_1:x_2,:);
    %Assembles a new, zoomed image

    r=bildzoom(:,:,1);
    g=bildzoom(:,:,2);
    b=bildzoom(:,:,3);
    %Extracts R, G and B from the image

    r_mean = mean(mean(r));
    g_mean = mean(mean(g));
    b_mean = mean(mean(b));
    %Averages

    rgb=[r_mean;g_mean;b_mean];
end
```