# Software for real time heart rate detection using a standard webcam

Maximillian F. Xavier
Departamento de Engenharia Eletrica
Faculdade de Tecnologia
Universidade de Brasilia
Brasilia, Brazil
maximillianfx@gmail.com

Otavio A. Dias
Departamento de Engenharia Eletrica
Faculdade de Tecnologia
Universidade de Brasilia
Brasilia, Brazil
otavioad@hotmail.com

Eduardo Peixoto
Departamento de Engenharia Eletrica
Faculdade de Tecnologia
Universidade de Brasilia
Brasilia, Brazil
eduardopeixoto@ieee.org

***Abstract*—Heart rate measurement is a valuable information in the description of an individual's condition. In some cases, measuring it without the need of physical contact is advisable. This work proposes a method capable of estimating a subject's heart rate using a standard camera based on subtle color variations on the skin caused by blood flow. This is a simple and low cost procedure with large applicability in many areas such as lie detection, medicine and surveillance. The proposed method is based on an established framework of face detection, filtering and peak detection in the FFT domain, adding to it a novel decision algorithm to make the heart rate estimation based on the past history of the FFT signal, its source code can be found in the team's GitHub repository. Results are good, within 5 bpm of the correct heart rate.**

***Keywords*-Heart rate, image processing, real time processing,**

## I. Introduction

The heart rate is a vital signal that can assist in a wide range of areas such as medical care, in which the monitoring may be important for prescription of suitable treatments or surveillance systems could also be improved to predict possible transgressions based on stress levels presented by an individual.

The blood flow variation over a certain region of the body performs a color change in the skin pigmentation that, in most cases, is invisible to the naked eye [1]. However, it has been shown that it is possible to capture these slight variations in the images captured by a common video camera and develop an algorithm in order to calculate the frequency at which these oscilations occur and estimate the heart rate of an individual [2] [3]. In this paper, we propose a real time algorithm to estimate the heart rate of a person. The algorithm's complexity is kept low so that it can be executed in a simple laptop computer.

Some diagnostic methods can be harmful for patients at risk or extremely vulnerable such as newborn babies or burn victms since they may involve touching or some sort of obstruction. Therefore, this method provides a safe and unharmful way of monitoring an individual's health condition and also reducing the amount of cabling needed since it involves only a simple camera and an average computer, which implies in an easy and cheap form of doing so [4].

Poh *et. al.* [2] dissertate about the possibility of capturing and measuring the heart rate of an individual through the color variation of their skin. A *region of interest* (ROI) is determined over the face by using the facial detection algorithm Viola-Jones [5], the average value of the pixels within the region provides valuable information when its channels are analyzed separately, which is called *independent component analysis* (ICA). The obtained results are then transferred to the frequency domain by using the *Fourier Transform*, enabling an evaluation of the peaks within a range of 40 to 240bpm. Those represent the maximum variability found and can lead to the expected frequency related to the desired heart rate.

Balakrishnan *et. al.* [3] propose a method for heart rate detection through involuntary head movement caused by blood flow. The subject has to be as statical as possible since any movement applied to an individual's face can interfere in the final results. Their work is compared to Poh's [2] as an attempt to find any relationship between color variation and head motion.

Wu *et. al.* [1] discourse about revealing temporal variations difficult or impossible to see with the naked eye. Their work is based on Eulerian Magnification, receiving as input a video and exaggerating subtle color changes.

Sandri *et. al.* [4] [6] proposed two methods that work in a similar manner as the method presented by Poh *et. al.* [2]. The first one differs in the application of an adaptive filter that multiplies the obtained signal in the frequency domain by a mask to mitigate regions with low probability of finding the expected heart rate, therefore, reducing the noise. The other algorithm proposed is an improvement of the first one to make it more robust to movements, employing a skin detector to eliminate pixels from the background of the region of interest, and a micro-region tracking in order to match regions in different frames.

This paper proposes a method to be implemented in an average computer connected to a standard webcam. It is based on an established framework of face detection, filtering and peak detection in the FFT domain, adding to it a novel decision algorithm to make the heart rate estimation [7] based on the past history of the FFT signal.

## II. Methodology

### A. Proposed framework

The proposed framework must be robust and fast enough so the real time data processing can be performed without any loss of information. The proposed framework is shown in Fig. 1, and detailed as follows:

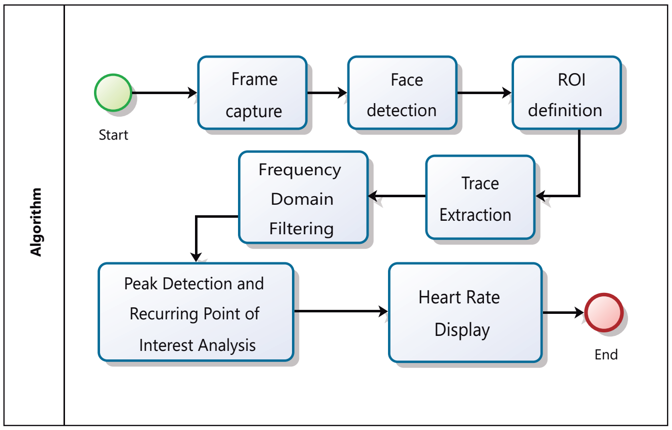



Fig. 1. Program workflow

*Frame capture*: The first stage consists of the acquisition of a frame through a webcam connected to the computer. As it is usual with most webcams, the captured frame is in the RGB colour format. Although the resolution of this webcam is not so important (a webcam with a $1280 \times 720$ resolution and 30 fps is used in the experiments), it is paramount that the webcam is able to capture the raw data, without any compression, as this would result in the loss of the color variations we are seeking. This is usually not a problem for webcams that are directly connected to the computer, but it must be observed for webcams connected through any kind of wired or wireless networks.

*Face Detection and ROI Definition*: The second stage consists of the facial detection over the acquired frame. As default, only one face is to be detected in this project, ignoring any other subjects appearing in the image. The third stage consists of defining a bounding box over the previously detected face. The ROI is defined as $h \times w$, where $h$ is the height of the bounding box of the detected face and $w$ is 55% of the width of this bounding box. The position of the ROI is centralized within the detected face. The resulting ROI is to be analyzed in the next stage. It was noted that, even though the computer used was capable of performing the Viola-Jones algorithm for every frame, the detected face varied slightly in position, changing the ROI of the algorithm due to the capture of involuntary movement and sometimes losing accuracy when detecting the facial region, making the heart rate estimation unstable. Therefore, the face detection algorithm is only performed for the first frame in each second (i.e., once every 30 frames). The ROI is kept the same for the remaining frames.

*Trace Extraction*: In the fourth stage, for every frame $n$, the ROI consists of a color image $h \times w$ with three channels, Red, Green and Blue. This image is normalized, removing the mean value for each channel, making the variance between samples unitary as follows:

$$\widehat{C}_i(n) = \frac{C_i(n) - \mu_i}{\sigma_i}$$

where i is for R, G and B, $\mu_i$ and $\sigma_i$ are the mean and standard deviation for each channel, respectively.

The signal we are looking for is embedded within this normalized image. Previous works [2] use Blind Source Separation techniques, such as Independent Component Analysis (ICA) in order to isolate the heart rate signal. The outcome of this blind source separation technique is three weights, so that each channel can be linearly combined in one channel. In a previous work by Sandri et. al [6], it was observed that the probability distribution of these weights are heavily skewed toward some values, namely: $\alpha_b = -0.4897$, $\alpha_r = -0.0244$ and $\alpha_g = 0.4956$. Therefore, in order to avoid using ICA and simplifying the algorithm for real time use, we define the trace simply as $T(n) = \widehat{G}(n) - \widehat{B}(n)$, where $\widehat{G}$ is the green channel normalized, $\widehat{B}$ is the blue channel normalized and $n$ is the frame number.

*Frequency Domain Filtering*: In the fifth stage, the signal of interest is the trace, $T(n)$. These algorithms are not applied for every new trace sample (i.e., every frame), but rather once every two seconds (i.e., once every 60 frames). The algorithm takes the trace values for the last 10 seconds and store it in a vector. It is then transformed to the frequency domain with a *Fast Fourier Transform - FFT* of 2048 points. It has been shown [8] that a simple derivative filter improves the signal-to-noise (SNR) ratio of the heart rate signal we are looking for. Therefore, we define a simple derivative filter in the frequency domain, with gain 0.2 up to 20 bpm, 1 from 150 bpm, and a linearly varying gain between these two frequencies.

*Peak Detection and Recurring Point of Interest Analysis*: This is the sixth stage, which performs the heart rate estimation by analyzing the existing peaks in the obtained signal after the previous stage. It is triggered only when the FFT is performed (i.e., every 2 seconds). The analyzed range is between 31 and 185 bpm, which corresponds to [0.52 3.10] Hz. The five magnitude peaks have their X-axis values (the bpm) stored and are considered as points of interest.

A common problem the algorithm faces is that, even after the application of the derivative filter, the trace image is noisy, and thus the trace FFT is noisy as well. It may happen that the heart rate signal we are looking for is masked by noise, which may have a higher magnitude than the heart rate signal itself, and therefore lead to wrong estimations. However, even in noisy conditions, the heart rate signal is still present. Therefore, we propose to use a different approach, which looks at the peaks not only in the current FFT, but in previous FFTs as well. The estimation will be the peak that appears most often with a high amplitude, not the highest peak at the current FFT. The idea is that the heart rate signal is always there, but the noise is random, appearing at different frequencies in different FFTs.

The algorithm looks at the 5 past FFTs. For each point of interest in the current FFT, the algorithm attempts to connect it to a point of interest in the previous FFTs, creating a *connection graph* for each of the five points of interest. A point of interest is considered to be connected if its X-axis values are within 5 bpm of each other. If no peak is found within 5 bpm, the connection graph is terminated. Once a connection graph is created for each point of interest, the algorithm computes a score. The score is simply a sum of weights, where the weight is assigned according to the relative magnitude of that peak in each FFT. The highest magnitude peak receives a score of 5, the second receives a score of 4, and so on until the fifth highest, which receives a score of 1. The estimation is considered as the point of interest that has received the highest score, even if it is not the highest peak in the current FFT. Fig. 2 shows the basic functioning of the procedure.

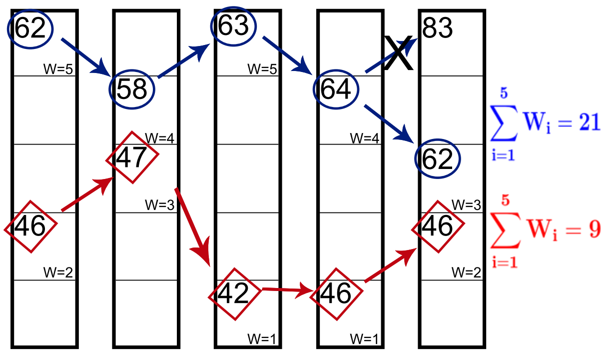



Fig. 2. Analysis of the points of interest

*Heart Rate Display*: The last stage consists of applying a moving average filter of size 3. That is, the result displayed on the screen is an average value between the last 3 estimations.

### *B. Software implementation*

The software [7] is implementated on a Windows environment using C++ progamming language aided by two libraries, those being OpenCV [9] and FFTW3 [10]. The OpenCV library provides the necessary resources to the facial detection algorithm and the frame capture through the webcam connected to a computer. The FFTW3 library is used because of the *FFT* algorithm it offers, as well as to manipulate complex numbers when needed. Although it was not tested here, the software developed does not rely on any OS specific functionality, and thus can be ported to a Linux environment.

## III. Experiments and Results

To validate the implemented software, four experiments were performed, which is shown in Fig. 3. Participants were seated at a table in front of a webcam attached to a computer, wearing a finger oximeter. Executions of 60 seconds were conducted and the results were analyzed in real time as they were given by the software and compared to the bpm shown by the oximeter. The experiments were performed using a standard laptop computer Asus with 6Gb of RAM, a Intel core i5-3317u processor with 1.7Ghz and an *onboard* graphics card. The webcam used is a 3 Megapixels Logitech recording at 30 fps.

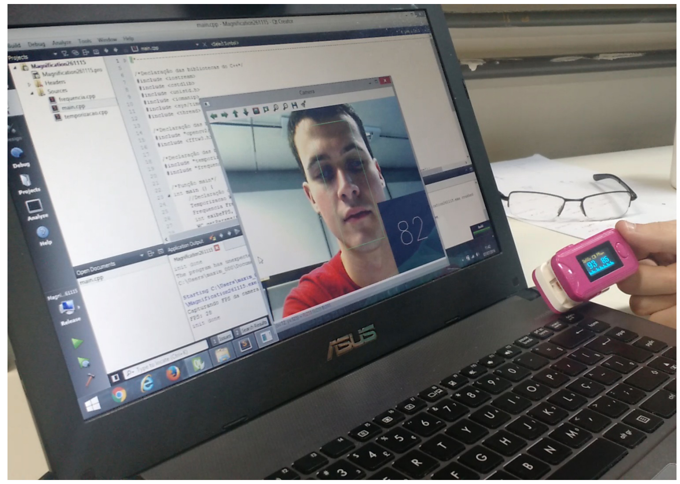



Fig. 3. Running experiment aided by an oximeter

The experiments were conducted with participants between 20 and 22 years old of different genders and races. Two experiments were performed with 22 year old white individuals. Subject 1 demonstrated an 80% efficiency result during execution, considering a range of ±5bpm while Subject 2 demonstrated an efficiency result of 43% considering the same bpm range. Fig. 4 shows the graphics of the obtained results, comparing data given by the oximeter along with those from the software.

Two other experiments were conducted with a black male subject and a brown female subject. The first one, named Subject 3, was 21 years old and the obtained results showed and efficiency of 80% during execution time. Subject 4 was a 20 years old female participant, providing an efficiency result of 60%. Fig. 5 shows the graphics of the obtained results, comparing data just like the previous experiments.

For this experiment, an efficiency comparison was made between using ±5bpm, ±8bpm and ±12bpm ranges, the obtained results are shown in table I.

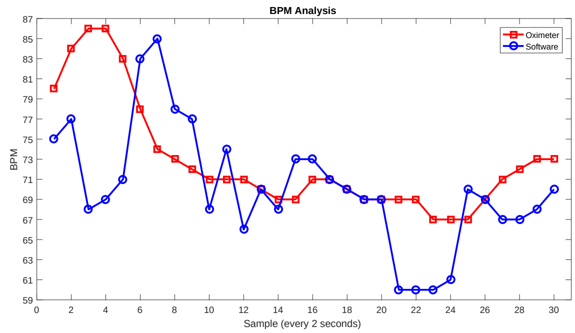
The research provided a determinant factor responsible for interfering in the estimations which is the luminosity of a room, which was considered a primary source of noise.

TABLE I
Efficiency Comparison

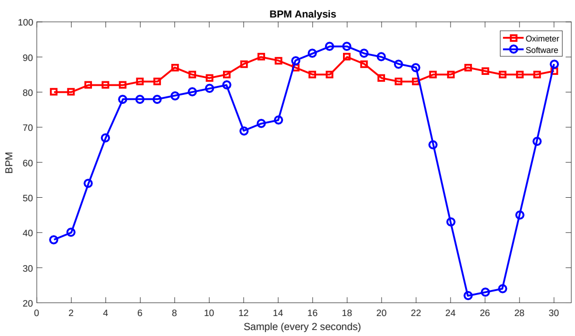
| | **Accuracy (%)** | | | |
|---|---|---|---|---|
| **Subject** | **1** | **2** | **3** | **4** |
| **±5bpm** | 80 | 43 | 80 | 60 |
| **±8bpm** | 87 | 53 | 80 | 63 |
| **±12bpm** | 97 | 53 | 80 | 70 |

## IV. Conclusion and Future Works

This work describes, implements and analyzes an unharmful, simple and cheap method of obtaining an individual's heart

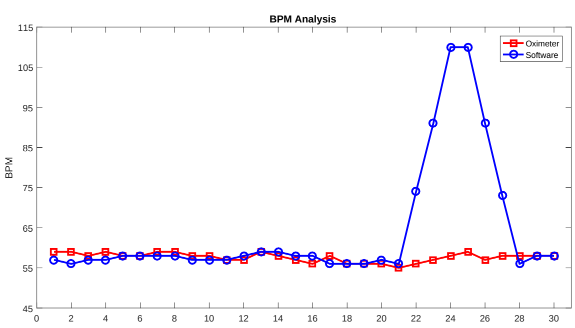



(a) Subject 1

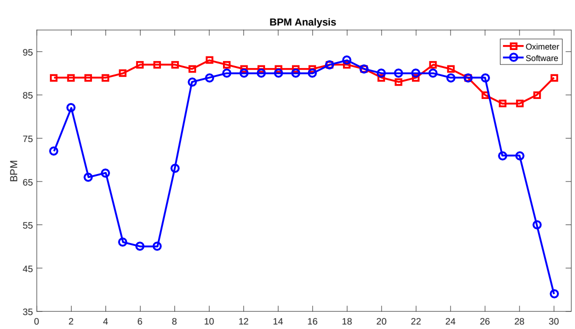



(b) Subject 2

Fig. 4. Experiments




(a) Subject 3




(b) Subject 4

Fig. 5. Experiments

rate in real time through a standard camera connected to an average laptop computer.The algorithm takes approximately 6 to 10 seconds to stabilize in cases of non abrupt movements and no relationship was found between estimation and skin color.

The obtained results confirmed to be as expected considering the machinery used for the experiments and the low complexity of the algorithm. As shown in table I, the $\pm$8bpm range can be considered the most suitable to the implemented software since it increased accuracy for 3 of 4 subjects. For future works it is advisable to improve the appliance of filters to minimize noise caused by ambient light and lower time needed to stabilize the estimation in cases of a moving subject.

## REFERENCES

[1] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, “Eulerian video magnification for revealing subtle changes in the world,” *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, vol. 31, pp. 651–658, 2012.

[2] M.-Z. Poh, D. J. McDuff, and R. W. Picard, “Non-contact, automated cardiac pulse measurements using video imaging and blind source separation,” *Optics Express*, vol. 18, pp. 10 762–10 774, 2010.

[3] G. Balakrishnan, F. Durand, and J. Guttag, “Detecting pulse from head motions in video,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference*, June 2013.

[4] G. L. Sandri, R. L. de Queiroz, and E. Peixoto, “Deteccao automatica de pulso cardiaco utilizando imagens de video,” in *XXXIII SIMPOSIO BRASILEIRO DE TELECOMUNICACOES - SBrT2015*, Juiz de Fora, MG, September 2015.

[5] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference*, vol. 1, 2001, pp. I–511–I–518.

[6] G. L. Sandri, “Automated non-contact heart rate measurement using conventional video cameras,” Master’s thesis, Universidade de Brasilia, Brasilia, Brazil, 2016.

[7] M. F. Xavier, O. A. Dias, and E. Peixoto. (2016) Shr. [Online]. Available: https://github.com/maximillianfx/SHR

[8] S. Xu, L. Sun, and G. K. Rohde, “Robust efcient estimation of heart rate pulse from video,” *Biomedical Optics Express*, vol. 5, p. 1124 1135, April 2014.

[9] A. Kaehler and G. Bradski, *Learning OpenCV*. O’Reilly Media, 2013.

[10] M. Frigo and S. G. Johnson. (2013) Fftw. [Online]. Available: http://www.fftw.org/