Autocorrelation-based, passive, non-contact photoplethysmography: computationally-efficient, noise-tolerant, extraction of heart rates from video

Chadwick Parrish, Kevin D. Donohue, and Henry Dietz;
Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky

Abstract

Photoplethysmography (PPG) is the detection of blood flow or pressure by optical means. The most common method involves direct skin-contact measurement of light from an LED. However, the small color changes in skin under normal lighting conditions, as recorded by conventional video, potentially allow passive, non-contact, PPG. Eulerian Video Magnification (EVM) was used to demonstrate that small color changes in a subject's face can be amplified to make them visible to a human observer. A variety of methods have been applied to extract heart rate from video.
The signal obtained by PPG is not a simple sinusoid, but has a relatively complex structure, which in video is degraded by ambient lighting variations, motion, noise, and a low sampling rate. Although EVM and many other analysis methods in the literature essentially operate in the frequency domain, fitting the video data to their model requires extensive preprocessing. In this paper a time-based autocorrelation method is applied directly to the video signal that exhibits superior noise rejection and resolution for detecting quasi-periodic waveforms. The method described in the current work avoids both the preprocessing computational cost and the potential signal distortions.

Introduction & Background

Photoplethysmography techniques recover a subject's heart rate by capturing small color changes seen through the skin. The mechanism behind these color changes are the capillaries expanding and contracting, which varies light reflections detected by the video signal. This signal is typically corrupted by noise, which comes from several sources, including lighting, motion, and quantization.

The goal of the current work is to be able to take photoplethysmographic data from a subject and reasonably extract the subject's heart rate, while minimizing the computational complexity of the algorithm and without amplification. As there are many things that could introduce noise into the sampled optical signal, it is also desirable to make the process as noise tolerant as possible. In order to minimize the effect of noise on the output result, autocorrelation was chosen for the processing, instead of the Fast Fourier Transform (FFT) that is used in much of the literature. There was only one work found which utilized short-term autocorrelation, by Das et. all[1]. In this work, short-term autocorrelation was used to obtain results with virtually no error for contact photoplethysmography.

Frequency domain processing was found to be the oldest and most common method for processing photoplethysmographic data. One of the works that seems to have inspired the popularity of this method was done by CSAIL at MIT, with the Eularian Video Magnification Project [2]. This project showed that there were recoverable signals that could be extracted through optical means, even though they are not visible to the naked eye. They also showed this magnification on subjects, stating that they could recover a subject's heart rate, then amplify it on the original video. To do this, they employed a series of complex techniques that end with a frequency-domain analysis. They use a filter from 0.8-1.0 Hz (48-60 BPM) on the video signal from their human subject recordings. These filter bounds are unrealistic, as the typical human heart rate typically ranges from 1-1.67 Hz (60-100 BPM), according to the American Heart Association [3]. Furthermore, the filter bounds should extend past 60-100 BPM, as it is desirable to measure as many different heart rates as possible. Later works have expanded the allowable frequency range, while still keeping some of the same frequency-domain processing techniques. Verkruysse et. all [4] provided another method of photoplethysmography, which includes using a band-pass filter (.8-6 Hz or 48-360 BPM) along with the FFT. There are three primary color channels for the recorded video; red, green, and blue. Each color channel will have a different Signal-to-Noise Ratio (SNR) for PPG signals, and some have implemented PPG using multiple. However, this work aims to be as computationally light as possible, so only the red color channel was used. In the work done by Verkruysse, the green and blue color channels were used. Multiple color channels were also used in the work by Blackford et. all [5], where averaged values for all three main color channels were used. Additionally, strong fill-in lighting was usedto provide optimal conditions for taking data from the subject's face. Spectral analysis was again used to be able to obtain the final results, as well as an independent component analysis (ICA) algorithm. ICA is a process in which a singular signal will be attempted to be separated into several non-Gaussian components. This process is explained in further detail in Tharwat's article [6].

Machine Learning has been growing in both usage and possibility in recent years, which has led some to be applying it to PPG for higher quality of results.Liu et all [7] used a Support Vector Machine (SVM) to extract their PPG signal. An SVM is a type of computationally expensive machine learning, and their particular one was trained to find faces, such that a PPG signal could be sampled off of each face. These signals would then be represented in the Hue Saturation Value (HSV) domain using the Hue (H) channel. After this signal was in the HSV domain, it was filtered and preprocessed using wavelet analysis. Afterwards, the heart rate value was extracted using the FFT. Their result was the frequency value with the highest peak amplitude.

A common theme throughout the literature is that algorithms will be overloaded with preprocessing and computational complexity for a seemingly small improvement in results. The current work aims to show that high computational complexity and heavy preprocessing demands can be reduce through the use of short-time autocorrection functions. Two different algorithms were written that sample data, either from a video file or on a live video on a Canon ELPH 180 Camera (this is possible on any CHDK supported camera, see [8] and [9]. Additionally, a data processing algorithm was written to extract the heart rate. All algorithms are explained in this paper, and the finalized source code will be made available on Aggregate.org.

VIRTAS: Video ROI Tiler and Sampler

In order to sample data for this work, a tool (dubbed VIRTAS) was developed to sample a given video for different color channels, and output a vector of time-stamped data to the data processing algorithm. Figure 1 shows the algorithm for VIRTAS. The input for VIRTAS is a video file and the frames per second number that the video was captured at. Initially, it performs a light check when the algorithm parses the arguments to ensure that are enough data points to operate on. If confirmed positive, it begins processing the given file. A child is then created, which uses FFMPEG (a multimedia framework [10]) to create a PPM (Portable Pixmap format, see http://netpbm.sourceforge.net/doc/ppm.html) for each frame, which get sent through a pipe to the sampling process. In order to set the ROI, the frame is divided into a nine by nine set of tiles. The tiles selected are set by default in VIRTAS, which are the center 25 tiles in the grid, or those from row three through seven, and column three through seven. The average of all red pixels within the ROI is calculated, and is then written to an array, and a corresponding time value written to a time array. When all frames are processed, the color array and time array both get printed in the console, such that they can be directly put into the Matlab data processing script.

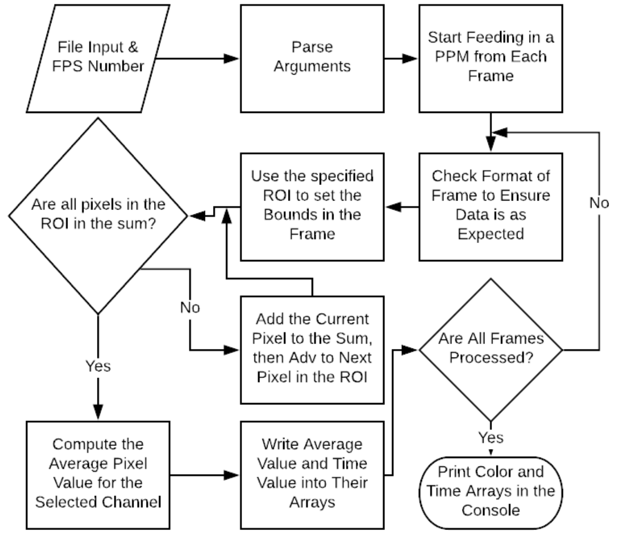
File Input & FPS Number
Parse Arguments
Start Feeding in a PPM from Each Frame
Are all pixels in the ROI in the sum?
Use the specified ROI to set the Bounds in the Frame
Check Format of Frame to Ensure Data is as Expected
No
No
Yes
Add the Current Pixel to the Sum, then Adv to Next Pixel in the ROI
Are All Frames Processed?
Yes
Compute the Average Pixel Value for the Selected Channel
Write Average Value and Time Value into Their Arrays
Print Color and Time Arrays in the Console

Figure 1. VIRTAS Flowchart

In-Camera Data Capture

The second sampling algorithm, written in Lua, has nearly identical functionality to VIRTAS. This script is run on a Canon Camera that has support for the Canon Hack Development Kit (CHDK). Details on Implementation can be found in [9], with a list of supported cameras located on the CHDK main website [8].

Figure 2 shows the algorithm flow chart for this in-camera data capture process. To run this script, the shutter button is pressed. The camera then takes in data through the optical sensor, and stores it in the liveview buffer. The liveview buffer is where data is stored that is shown on the rear screen. The buffer is then divided into a nine by nine grid, with each grid location being referred to as a tile. The ROI for this algorithm is the center five by five tiles, or those from columns three through seven, and rows three through seven. A spatial average for the red color channel is then calculated for each tile within the ROI. The average values then get summed up, and a time value is taken for that sample. These two values get written into the same position in a color array (the signal to be processed) and a time array (in case resampling is desired). This process happens 600 times, one sample per time for twenty-five samples per second. After the process concludes, the color array and time array are written to a file in the top directory of the camera's SD card.

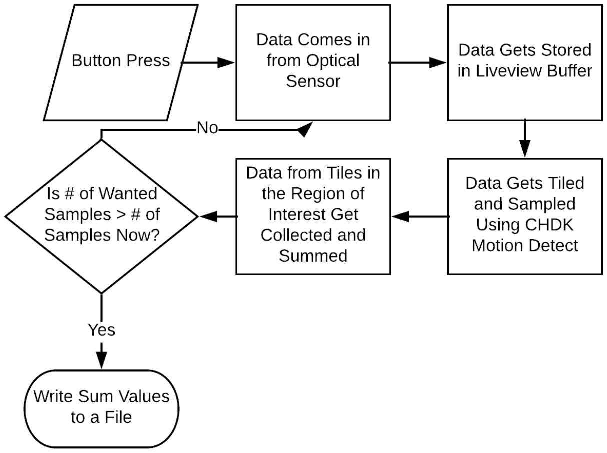
Button Press
Data Comes in from Optical Sensor
Data Gets Stored in Liveview Buffer
No
Is # of Wanted Samples > # of Samples Now?
Data from Tiles in the Region of Interest Get Collected and Summed
Data Gets Tiled and Sampled Using CHDK Motion Detect
Yes
Write Sum Values to a File

Figure 2. In-Camera Data Capture Flowchart

Data Processing

The traditional process for extracting a heart rate from photoplethysmographic data uses a band-pass filter/amplifier to isolate the desired frequency range corresponding to expected heart rates. Then, an FFT of the filtered data is performed and the heart rate frequency is determined by the largest peak amplitude in that range. However, in low SNR cases there will typically be many peaks generated by the noise and interference, and these can be greater than the peak generated by heart signal. In addition, since the heart signal is not a perfect sinusoid, its energy in the frequency domain is spread out over the harmonic frequencies (integer multiples of the fundamental), which lead to a lower dominant peak. The autocorrelation, however, transforms the signal and noise energies allowing for better separability, and better maintains the periodic signal power. The uncorrelated nature of system and quantization noises are all captured by the zero lag of the

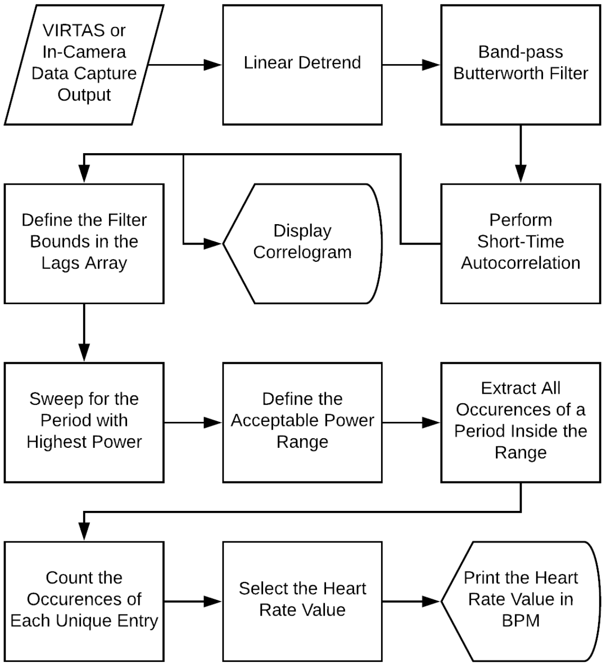
VIRTAS or In-Camera Data Capture Output
Linear Detrend
Band-pass Butterworth Filter
Define the Filter Bounds in the Lags Array
Display Correlogram
Perform Short-Time Autocorrelation
Sweep for the Period with Highest Power
Define the Acceptable Power Range
Extract All Occurences of a Period Inside the Range
Count the Occurences of Each Unique Entry
Select the Heart Rate Value
Print the Heart Rate Value in BPM

Figure 3. *Data Processing Flowchart*

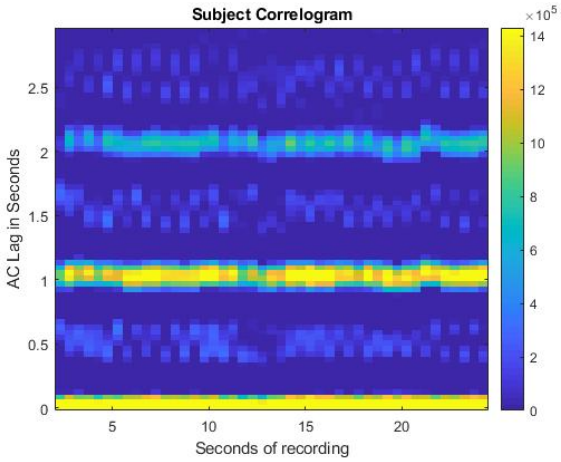

Figure 4. *High SNR Data, Taken From LEDs*

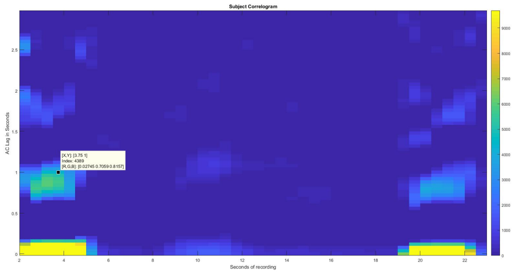

Figure 5. *Low SNR Data, Trial 2.2*

autocorrelation, and do not interfere with the amplitude at latter lags corresponding to periodic behavior. In the frequency domain this noise is spread out over all frequencies and interferes with the peaks in the heart rate ranges. In addition, the autocorrelation detects the periodic waveform by correlating/matching-up the beat waveform with the following beat, where correlations with full the waveform shapes, and not a narrow-band filtered version of it. As a matter of fact the more complex the shape is the more distinct the peaking will be at that associated lag. Thus, a much greater portion of the signal energy is used by the autocorrelation method than by the frequency domain method, which filter out the harmonic energy of the beating heart signal. The data processing algorithm developed for the current work calculates a series of short-time autocorrelations, which are then used to create a correlogram (analogous to the spectrogram). The correlogram is simply a display of a sequence of short-time autocorrelation amplitudes mapped into a color code and arranged sequentially. This creates an image from which the periods (if a periodic signal is present) over short-time interval can be observed. Figures 4 and 5 are examples of correlograms. The correlogram shown in Figure 4 is an example of an output for high SNR, as the data for this was from a red LED light in a dark room. There are three regions of the correlogram that describe this signal. Note at the zero lag (bottom of vertical axis), there is a yellow bar stretching over all time intervals, which is proportional to the total energy in the signal (it includes energy from both signal and noise, since it is the sum of the squared values of each sample). So all other peak amplitudes can be assessed relative to the total energy at the zero lag. The next important lag region is from .33 to 1.25 seconds. These periods correspond to frequencies of 3 Hz down to .8 Hz, which is the expected heart rate range. From this region, it can be determined whether or not a strong periodic signal is present as well as its period. Given the total signal plus noise energy is at the zero lag, a peak associated with a periodicity that is close to the amplitude at lag zeros, means there is very little noise energy present in the signal (i.e. it is dominated by periodic signal of interest). The third region of interest is located above the the 1.25 seconds lag, where there could be peaks at integer multiples of the signal period. These are not harmonics as seen in the frequency domain, where signal energy is dissipated over each one. These are simply multiples of the fundamental period over which the signal matches itself over multiples of the signal period. The additional peaks can occur when there is a very regular heart beat extended over multiple peridos with little change. For example if a heart beat is very regular at 1 second, there will be matches for lags at every 1 second, 2 seconds, 3 seconds and so forth. Figure 4 shows an example of this with peaking near 1 and 2 seconds. Since the short-time interval was less than 3 seconds another peak does not appear. During more realistic test cases, such as in ambient lighting outdoors, or in AC lighting indoors, the SNR is usually too low for peaking at multiple periods to be visually obvious, as is shown in Figure 5.

Testing

In order to verify the functionality and the accuracy of the current work, it was tested under a variety of different conditions, with each test using the same procedure. Preliminarily, the camera was placed 46 inches feet away from the subject, with the subject's nose in the center of the frame. The camera's was zoomed in at its maximum optical zoom, with the digital zoom being disabled. Figure 6 shows how a subject's face was oriented in the camera frame during data collection.

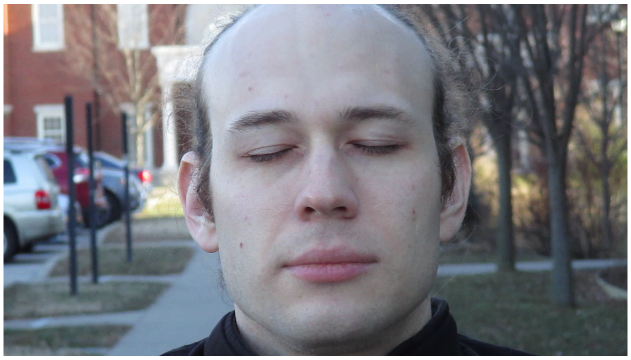

Figure 6. Facial Layout Within a Frame

The following procedure was used to take all data which utilized VIRTAS:

1. Verify that the subject's face is in the center of the frame, and that there is as much of the screen covered by the face as possible. Ideally, the left and right edges of the face should still be visible, to ensure that the ideal facial regions for taking data are still present in the frame.
2. Take the video. This should ideally be at least 10 seconds, as this should provide enough samples for credible results.
3. Run the video through VIRTAS, as specified previously.
4. Take the output color data and time data from VIRTAS, and input this into the Data Processing Matlab code, along with the sampling frequency that was used in VIRTAS.
5. Run the script. This code will output a correlogram, and the BPM value (within the filter bounds) with the highest energy will be printed in the console.

There was also a second testing procedure that was developed for trials using the In-Camera Data Capture algorithm, such that there could be trials taken with it as well. For this algorithm, the subject was exactly three feet away from the camera. This procedure was used to take all data which utilized the in-camera algorithm:

1. Verify that the subject's face is in the center of the frame, and that there is as much of the screen covered by the face as possible. Ideally, the left and right edges of the face should still be visible, to ensure that the ideal facial regions for taking data are still present in the frame. For this algorithm, three feet away from the face is the distance that was selected, which allows for good coverage at the ELPH 180's maximum optical zoom.
2. Run the CHDK program. The screen will say show that the program is started, and will also indicate when the program is finished.
3. Remove the camera's SD card, and connect it to a computer. The file ”heartbeat”, located in the top directory of the SD card, will have the data on it.
4. Insert the data into the data processing script. The first set of numbers within the square brackets will have the color data in it, and the second will have the time data. While the called function requires the time data array as an input, it will only be used if resampling is desired. The results shown below were not resampled.
5. Run the script. This code will then output a correlogram, and the BPM value (within the filter bounds) with the highest energy will be printed on the console.

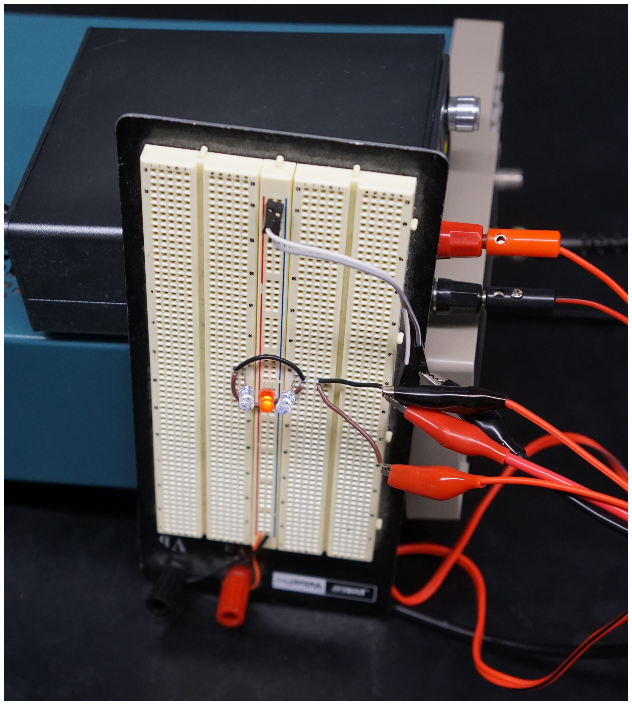

Figure 7. Setup of Simulated Data Using LEDs

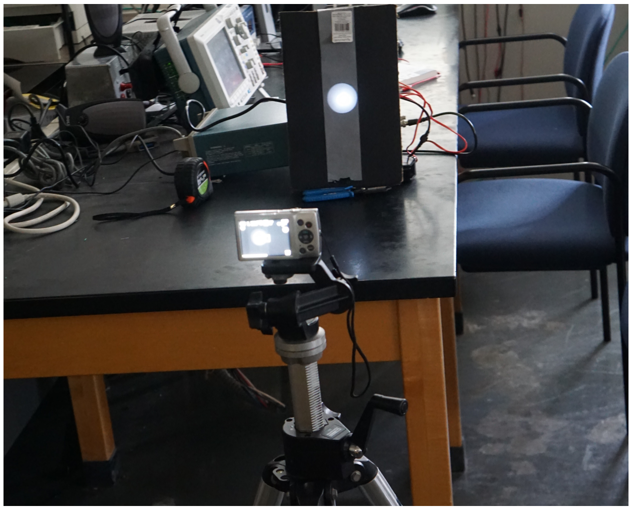

Figure 8. Testing Setup for Simulated Data Trials

This procedure was used in multiple different settings to provide a wide range of results. The first test was done on a simple red LED light that was driven from a function generator, all located in a dark room. The oscillation frequency was driven using a function generator that was accurate to at least the first decimal place, which is the same level of precision as the results from the script that it would be compared to. There were two more rounds of testing that were done, with both being taken outdoors. The first set of data utilized VIRTAS for data acquisition, as videos were taken on the Canon PowerShot ELPH 180, then run through VIRTAS and the data processing code. The final set of data was taken using the in-camera data capture process. This data was then loaded directly into the data processing code. The results for these trials are shown and further analyzed in the next section.

Results

In this section, final results from the data processing algorithm are presented. Data from both data capture algorithms were used, such that there could be a definitive Absolute Mean Error for each when paired with the data processing algorithm. Table 1 shows the results for the data sampled from videos using VIRTAS, accompanied by the duration (Dur.) of the taken video. Table 2 shows the results for the data sampled using the in-camera data capture algorithm. For both sets, a cellphone heart rate monitor was used to acquire the reference (Ref.) BPM, and the estimated (Est.) BPM is the output number from the data capture algorithm. In both sets of trials, subject 1 is a Caucasian male, and subject 2 is an African-American female.

The results for the proof-of-concept data are shown in Figure 4. This data has a very high SNR, with the input frequency being 1 Hz, and the measured BPM being 62.5 BPM, which is just one quantization level away from 60.0 BPM. This trial has virtually the best results possible, as the correlogram looked exactly as it was predicted to. Figures 7 shows the setup of the LED lights, and Figure 8 shows the setup of the camera with relation to the lights. Although these figures show lights on in the room, when data was taken the lights were turned off and the blinds shut, such that there was no light that didn't originate from the LEDs.

Trial	Subject	Dur. (seconds)	Est. BPM	Ref. BPM	Abs. Error (BPM)
1.1	1	20	136.4	80	56.4
1.2	1	20	107.1	80	27.1
1.3	1	20	64.9	80	15.1
1.4	1	30	68.0	80	12.0
1.5	1	30	62.1	80	17.9
1.6	1	30	95.2	80	15.2
1.7	2	25	79.4	81	1.6
1.8	2	16	79.4	81	1.6
1.9	2	29	75.2	81	5.8
1.10	2	33	71.4	81	9.6
1.11	2	32	71.4	81	9.6

Table 1: VIRTAS Data Capture Results

For the set of data which utilized VIRTAS, the absolute mean error for subject one was 24.0 BPM, and the absolute mean error for subject two was 5.6 BPM. The total absolute mean error for both subjects was 15.6 BPM. These results were unexpected, as traditionally photoplethysmography is more accurate on those with darker skin tones. Potential sources of error in this set are that cellphone measurement of Subject One's heart rate was too high, which would have created more error in trials 1.3, 1.4, and 1.5. It is very likely that there was an octave error, or error that makes it display a frequency that is 2^n higher than the true one, in trial 1.1. This is known because the subject had not undergone any aerobic exercise close to the trials, and has a typical resting heart rate of 68-72 BPM. Another major difference in the trials between subject one and subject two were the lighting conditions outdoors when the subjects were available. Subject one's data was taken in the morning, when there was no direct sunlight on his face. Subject two's data was taken in the afternoon, when there was more ambient sunlight, although not directly shining on her face. For the set of data which utilized the in-camera data capture algorithm, the absolute mean error for subject one was 2.72 BPM, and the absolute mean error for subject two was 14.04 BPM. It also appears that trial 2.8 had an octave error, similar to trial 1.1.

Trial	Subject	Est. BPM	Ref. BPM	Abs. Error (BPM)
2.1	1	75	68	7.0
2.2	1	68.2	68	0.2
2.3	1	65.2	68	2.8
2.4	1	71.4	68	3.4
2.5	1	68.2	68	0.2
2.6	2	68.2	76	7.8
2.7	2	68.2	76	7.8
2.8	2	125	76	49
2.9	2	75	76	1.0
2.10	2	71.4	76	4.6

Table 2: In-Camera Data Capture Results

Overall, the absolute mean error was 8.38 BPM. When there is fill-in lighting on the subject's face, the results from both algorithms seem comparable to other remote works, especially for the amount of processing being done to get the results. Sources of error and challenges for getting the most accurate results are listed in the difficulties section.

Difficulties

There are many difficulties that are encountered when trying to develop and test algorithms for remote photoplethysmography. Some difficulties were already documented, such as very low SNR, the non-uniformity of heart rates, and the need for active lighting. There were also difficulties that were discovered during the development of this work, such as the presence of signal energy and low-frequency noise issues.

Low SNR is an issue that has been prevalent for as long as PPG has existed. Active lighting has been a solution to this in many works, but it is more desirable to create a workable method that can be entirely passive. Active lighting itself could cause issues, for if the active light source uses pulse width modulation (PWM) there will be additional periodic noise introduced which could hinder the ability of any PPG algorithm to recover the true heart rate. Another issue which makes PPG signal recovery more difficult is that heart rates change over time. This is likely a source of error in the documented trials, as the reference heart rate was taken at the beginning of the trial set. Heart rates will decline as people rest after being active (walking, pushing a chair, or most activity), but the heart rate of some will increase when a camera is pointed at them. The non-uniformity of heart rate signals doesn't just extend to the BPM values, as the signals themselves can vary. Heart rate signals, while periodic, are not sinusoidal. This makes recovery harder, but especially in cases where the heart rate is assumed to be sinusoidal. In cases where the SNR is low, and the number of bits used to represent the signal is also low, then random noise or quantization error could influence the results such that a harmonic would have the highest peak amplitude. The ELPH 180 has 8 bits that it uses for storing its color values, and while this would ideally be 16 or 32, 8 is still usable.

Lighting conditions are also an issue when recovering a PPG signal. This work demonstrated this in Trials 1.1-1.6, as those results were in dimmer lighting conditions than the others, as the light was coming from the opposite direction that it did in the other trials. The current work utilizes reflectance mode PPG, where the sampled signal is light reflected off of the subject's face. This signal is degraded by quantization error, but the signal will have a larger amplitude if there is more light being reflected off of the subject's face. This makes signal recovery in dark rooms or dark shaded areas difficult. A larger number of bits would help this issue, but would likely still present a challenge. It is also worth noting that sampling the green color channel may provide better results as well in this case, as there are 2x as many green pixels in the Bayer Matrix as there are red or blue. The other work proposed by Dietz et all [9] has a good discussion of this, as this other work did not use the red color channel.

Low-Frequency noise and octave errors were two issues that presented challenges in the selection of the filter bounds. It was found that when a pure low-pass filter was implemented, there were at least two high noise peaks below .8 Hz, which were approximately located at .2 Hz and .5 Hz. These noise peaks were in some cases higher than the signal, or comparable in peak amplitude. This led to the selection of .8 Hz as the lower bound, leading to the lowest recoverable heart rate being 48 BPM. In an ideal case, this would be .6 Hz such that 40 BPM would be included in the recoverable range. Even though it is rare that a subject will have this heart range, it would be ideal to find a method such that 40 BPM is both recoverable and not contaminated by noise. Octave errors were a determining factor in setting the higher frequency bound. A simple solution to this would be to define a probability density function (PDF) to the filter bounds, such that the normal resting heart rate would be more likely to happen than 2x or 3x normal resting heart rate. If the correlogram was multiplied by that PDF, then octave errors would be much less likely. An obvious shortcoming of this approach is that it heavily biases the data, and makes it significantly harder to recover frequencies that are located outside of that usual heart rate range (outside of 60-100 BPM). To limit the number of octaves, and as a heart rate of greater than 180 BPM is very uncommon, the top filter bound of 3 Hz was chosen. The final source of difficulty that arose in the development and testing of the current work was the inconsistency of energy in the correlogram. As was noted previously, Figure 5 shows that inconsistency. Even with lighting conditions being consistent to the human eye, the energy changes throughout the duration of the data sequence. Furthermore, there is a loss of signal energy when using VIRTAS as opposed to the in-camera data capture algorithm. The energy from data in VIRTAS is typically at least two orders of magnitude lower than that of the energy of the data from the in-camera data capture process. Some of this loss comes from the extra stage of compression that the video undergoes when it is being saved and written, whereas that doesn't happen when sampling from the liveview feed.

Conclusion

The current work proposes a method of photoplethysmography that is computationally inexpensive, such that it performs a minimal amount of processing and has no amplification. It has been shown that it is able to recover results from a high SNR case, the LED array, and from a low SNR case, off of the faces of two different subjects with very different skin tones. This work has all be done with no additional or focused lighting used, such that this data could be taken by any user in an office complex or park. The future plans for this work, and that of COIMG-146, are to further develop the algorithms for real-time embedded applications. These efforts will first be directed at creating a more accurate and stable embedded process, then to optimize the device in real-time. A facial tracking algorithm is also in development such that a higher level of motion denoising can occur. After these are all implemented, the final planned expansion of this work is to develop a version which can do this on multiple subjects simultaneously.

Acknowledgments

The authors would like to acknowledge Paul S. Eberhart for his assistance in the testing and revision stages of this work, as well as Grace Oparinde for her assistance in the testing stage of this work.

References

[1] S. Das, S. Pal, and M. Mitra. Real time heart rate detection from ppg signal in noisy environment. In *2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI)*, pages 70–73, Oct 2016.

[2] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012.

[3] American Heart Association. All about heart rate (pulse). https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse, Jul 2015.

[4] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Opt. Express*, 16(26):21434–21445, Dec 2008.

[5] Ethan B. Blackford, Justin R. Estepp, Alyssa M. Piasecki, Margaret A. Bowers, and Samantha L. Klosterman. Long-range non-contact imaging photoplethysmography: cardiac pulse wave sensing at a distance. volume 9715, pages 9715 – 9715 – 17, 2016.

[6] Alaa Tharwat. Independent component analysis: An introduction. *Applied Computing and Informatics*, 2018.

[7] LingLing Liu, Yuejin Zhao, Lingqin Kong, Ming Liu, Liquan Dong, Feilong Ma, and Zongguang Pang. Robust real-time heart rate prediction for multiple subjects from facial video using compressive tracking and support vector machine. *Journal of Medical Imaging*, 5:5 – 5 – 6, 2018.

[8] Chdk wiki. https://chdk.wikia.com/wiki/CHDK.

[9] Henry Dietz, Chadwick Parrish, and Kevin Donohue. Self-contained, passive, non-contact photoplethysmography: Real-time extraction of heart rates from live video within a canon powershot, coimg-146, 2019.

[10] Fabrice Bellard. Ffmpeg. https://www.ffmpeg.org/.