Real-time Analysis of Photoplethysmogram and Electrocardiogram for Assessment of Physiological Correlates of Attention

Bryce Landen Gautney

University of Alabama in Huntsville

Follow this and additional works at: https://louis.uah.edu/honors-capstones

Recommended Citation
Gautney, Bryce Landen, "Real-time Analysis of Photoplethysmogram and Electrocardiogram for Assessment of Physiological Correlates of Attention" (2024). Honors Capstone Projects and Theses. 887. https://louis.uah.edu/honors-capstones/887

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.
Real-time Analysis of Photoplethysmogram and Electrocardiogram for Assessment of Physiological Correlates of Attention

by

Bryce Landen Gautney

An Honors Capstone
submitted in partial fulfillment of the requirements
for the Honors Diploma
to
The Honors College
of
The University of Alabama in Huntsville
May 3, 2024
Honors Capstone Project Director: Dr. Emil Jovanov

S tudent (signature) D ate

Project Director (signature) D ate

Department Chair (signature) D ate

Honors College Dean (signature) D ate
Honors Thesis Copyright Permission

This form must be signed by the student and submitted with the final manuscript.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Bryce Gautney
Student Name (printed)

Bryce Gautney
Student Signature

May 5, 2024
Date
Abstract:

Improvement of human attention and cognitive function can provide many benefits to psychological tasks. This project aims to create a non-invasive method of estimating heuristics for attention and cognitive effort via the use of Photoplethysmogram (PPG) and Electrocardiogram (ECG) devices. These two non-invasive devices which monitor heart activity can be used to observe and estimate several characteristics of the physiological state of a user, which can be used for monitoring and to provide feedback to the user via the Attention Monitoring and Cognitive Enhancement System (ACES).

Thesis:

Research has shown that PPG and ECG signals can be combined in analysis to estimate several factors that are indicative of a user’s current stress and cognitive effort. Hosseini and Lowe were able to use these signals and their characteristics to create a convolutional neural network that was able to estimate blood pressure, a common metric of stress, within a small and acceptable margin of error. This is just one of the ways these signals can be analyzed when combined with each other, with many more that can be explored.

Given the importance of these two symbols combined, it is essential to have a system that can not only record these measurements but also calculate several defining characteristics of the signals in real time so that these process can be applied while a user is actively connected to the system rather than as post-operational analysis. The key characteristics I have discovered for this analysis are the following: the difference in time between an ECG peak and the following PPG pulse’s middle known as Pulse Travel Time (PTT), the difference in time between subsequent peaks of ECG pulses which is known as the R-Wave Interval (RR), the relative magnitude of an ECG pulse, the height and width of a PPG pulse, and the difference in time between the middle of the previous PPG pulse and the peak of the subsequent dicrotic notch known as Dicrotic Notch Latency (TD).

These following plots demonstrate the characteristics that are being identified based on the plot of the raw ECG and PPG data. The blue line is the ECG signal and the red line is the PPG signal. In the first graph the PPG signal has been scaled up and shifted vertically for better presentation of information.
To perform the analysis required to gather these data points in real time, I implemented a real-time filter developed by Svitzky and Golay on incoming data that was able to give the derivative of a smoothed version of the raw signal. This filter is extremely fast and efficient in calculating this derivative, requiring only a window size of 5 and with only integer coefficients. Through the use of this filter I was able to locate the peaks, valleys, and middle of the PPG signal in real time with only a delay of two samples, which at the current sample rate is only a total delay of 16ms, which can be accounted for when recording the data.

The data gathering, filtering, and calculations are being conducted on a HealthyPi v5 board with its provided PPG finger sensor and ECG electrodes. This device significantly expedited development process as it was already
designed to gather the raw data that this project uses, so it was only the filtering and analysis that needed to be constructed as additional programming. The algorithm has been developed both in C for real-time operation on the HealthyPi board as well as in Python for post-operational analysis on data that was gathered before the implementation of the C program.

The data filtering and gathering is fully implemented onto the HealthyPi. The different values are all sent over USB communication to a receiver program once every heartbeat. The RR values recorded can be used to estimate heartrate in real time as they are inversely proportional to each other. PTT and TD are both able to be combined to estimate blood pressure as Hosseini and Lowe were able to accomplish with their neural network. These characteristics can be used via a researcher or neural network created by a researcher in tandem with the other recorded characteristics to analyze a patient’s physiological state in ways that are outside the scope of this project and my research. The following code is the operation of the algorithm on the HealthyPi board.

```plaintext
sendFlag = false;

ecgWindow[0] = ecgWindow[1];
ecgWindow[1] = ecgWindow[2];
ecgWindow[2] = ecgWindow[3];
ecgWindow[3] = ecgWindow[4];
ecgWindow[4] = ecg_sample;

//Finds the peaks of ECG signal.
    ecgWindow[3] >= 20000000) {
    RR = 8*((sampleCount - 1) - ecgPeak);
    ecgPeak = sampleCount - 1;
    ECMG = ecgWindow[3] - ecgWindow[0];
}

ppgWindow[0] = ppgWindow[1];
ppgWindow[1] = ppgWindow[2];
ppgWindow[2] = ppgWindow[3];
ppgWindow[3] = ppgWindow[4];
ppgWindow[4] = ppg_sample;

//This filter finds the derivative of the smoothed curve of the PPG signal for t = -2
//As such filteredWindow[2] refers to the derivative at the same time as ppgWindow[2] not ppgWindow[4]
    filteredWindow[0] = filteredWindow[1];
    filteredWindow[1] = filteredWindow[2];
         ppgWindow[3] + 2*ppgWindow[4];
```
//Finds local minima of certain magnitude for the derivative to
determine where the PPG pulse's slope is the most negative.
//This gives an estimate for the middle of the PPG pulse.
if (filteredWindow[1] <= filteredWindow[0] && filteredWindow[1] <=
dFlag = true;
pulseMid = sampleCount - 3;
PTT = 8*((sampleCount - 3) - ecgPeak);
}
//Find where the derivative goes from negative to positive. This
correlates to the valleys in the PPG signal.
else if (filteredWindow[1] <= 0 && filteredWindow[2] >= 0){
  //If the flag is set then this is the pulse minimum value.
  if (dFlag){
    PW = 8*((sampleCount - 2) - pulseStart);
    PH = pulseMagMax - ppgWindow[2];
  }
}
//Find where the derivative goes from postivie to negative. This
correlates to the peaks in the PPG signal.
else if (filteredWindow[1] >= 0 && filteredWindow[2] <= 0){
  //If the flag is set then this is the dicrotic notch peak.
  if (dFlag){
    dFlag = false;
    TD = 8*((sampleCount - 2) - pulseMid);
    sendFlag = true;
  }
  //If the flag is not set then this is the pulse maximum value.
  else{
    pulseStart = sampleCount-2;
pulseMagMax = ppgWindow[2];
  }
}

This is called every 8ms for a sample rate of 125Hz on the PPG and ECG, keeping a window of previous
values for analysis, performing the filtering required to find the derivative of the PPG signal, and then executing the
analysis of that derivative to determine the location of key points used to determine the output values. All variables
besides flags and the sampleCount are 32-bit signed integers, which are booleans and a 32-bit unsigned integer
respectively. Afterwards if sendFlag is set then the data is compiled into a packet and sent via USB serial.
This sample of recorded data shows the format that is output by a Python script reading the values sent over USB serial communication from the board. The data packets are sent as a 5 byte header, a 28 byte data segment, and a two byte footer. Each 4 bytes of the data segment comprises one of the following 32-bit numbers, which are broken down into bytes for transmission over the wire. Information on the header and footer can be

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Segment</th>
<th>Footer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tms, RR, ECGM, PTT, PW, PH, TD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1080, 600, 58720258, 496, 120, 51692, 232</td>
<td>1368, 600, 58720258, 280, 528, 133241, 136</td>
<td>1648, 56, 66519043, 464, 664, 258463, 232</td>
</tr>
<tr>
<td>1960, 56, 66519043, 184, 1064, 171913, 216</td>
<td>2208, 616, 68485119, 432, 1256, 215978, 216</td>
<td>2784, 200, 63504387, 0, 128, 53709, 216</td>
</tr>
<tr>
<td>3344, 56, 67502082, -16, 120, 52361, 216</td>
<td>3880, 64, 59506690, 16, 120, 50475, 224</td>
<td>4424, 344, 65929219, 152, 112, 50379, 224</td>
</tr>
<tr>
<td>4992, 376, 66191359, 192, 104, 48892, 224</td>
<td>5608, 616, 75169791, 424, 104, 50474, 232</td>
<td>6336, 728, 67633152, 528, 112, 53798, 240</td>
</tr>
<tr>
<td>7104, 760, 63111171, 568, 112, 56190, 240</td>
<td>7792, 696, 69926909, 512, 112, 47952, 224</td>
<td>8424, 632, 64618497, 456, 120, 52281, 216</td>
</tr>
<tr>
<td>9016, 592, 64552961, 416, 120, 51710, 216</td>
<td>9584, 560, 71237632, 376, 112, 50381, 232</td>
<td>10136, 560, 70713344, 368, 112, 47277, 232</td>
</tr>
<tr>
<td>10736, 600, 74579970, 416, 112, 47294, 224</td>
<td>11408, 672, 75497472, 488, 104, 45936, 224</td>
<td>12104, 696, 55836669, 520, 104, 43910, 216</td>
</tr>
<tr>
<td>12776, 672, 70057984, 488, 104, 40317, 224</td>
<td>13392, 600, 55181312, 432, 112, 37887, 224</td>
<td>13936, 544, 41746434, 384, 112, 34404, 216</td>
</tr>
<tr>
<td>16960, 824, 37421058, 648, 88, 28760, 248</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The program has been validated by recording its data alongside raw recordings of the PPG and ECG signals and manually analyzing beats via Matlab plotting to determine any errors generated by the program. The following is the validation process for a single Matlab plot. This is done using a different C and Python program that outputs the sample and characteristic values for every sample rather than every heartbeat.

1) Zoom in on two subsequent PPG pulses.

2) Locate the absolute peak of the first pulse, absolute valley of the first pulse, and the dicrotic notch following that valley.
3) Estimate the midpoint of the first PPG pulse.

4) Calculate the characteristics that are solely based on PPG signal and compare them to what the program listed at the time of the dicrotic peak + two sample rates to account for the delay in processing caused by the filter.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Manual Observation</th>
<th>Program Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>PW</td>
<td>88</td>
<td>96</td>
</tr>
<tr>
<td>PH</td>
<td>101000</td>
<td>100959</td>
</tr>
<tr>
<td>TD</td>
<td>208</td>
<td>208</td>
</tr>
</tbody>
</table>

As you can see first two values are slightly off while the TD is exactly as the program output. This is likely to me selecting a peak value one sample off from what the algorithm estimated, but is easily within acceptable error.

5) Unselect all your points except the midpoint and zoom out to select a the peak and a baseline value from the ECG spike before that PPG pulse as well as the peak from the previous ECG pulse before that. Use the program’s values from the same timestamp.
6) Repeat your analysis as done earlier.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Manual Observation</th>
<th>Program Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR</td>
<td>456</td>
<td>464</td>
</tr>
<tr>
<td>ECMG</td>
<td>58459000</td>
<td>57671681</td>
</tr>
<tr>
<td>PTT</td>
<td>264</td>
<td>264</td>
</tr>
</tbody>
</table>

The values are once again extremely similar with any error likely due to slightly altered point selection. The magnitude is within a 1.35% error.

7) This process can be repeated, and was during testing, to ensure the program’s validity.

These results show that this program and hardware are a capable method of calculating these essential characteristics from a user in order to provide non-invasive monitoring of their physiological state in real-time.

I did not accomplish as much as I would have liked throughout this project. I would’ve liked to have actually implemented the blood pressure and cognitive effort estimations in my programs, but I was unable to get a strong enough grasp on both the artificial intelligence and the biological aspects required to develop such an algorithm. This project can be used as the cornerstone of such a future project as it provides all the required datapoints that can be analyzed going forward. It also will be usable for refinement of the ACES which was the senior design project this project was a component of and which seeks to use this data to provide stimulation to a user to improve their performance in cogntivite tasks.

References:


Smoothing and Differentiation of Data by Simplified Least Squares Procedures.

Abraham, Savitzky and M. J. E. Golay
Analytical Chemistry 1964 36 (8), 1627-1639
DOI: 10.1021/ac60214a047


Tables and Graphs:

A Matlab implementation of the algorithm used to determine the peaks of PPG signals:

A sample of recorded values from the program running over the course of a couple minutes.