A5- Photoplethysmography (PPG)

CS390MB Android Assignment

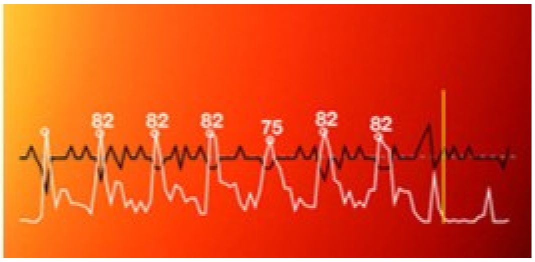
In this assignment we will be using the camera and flash on the smartphone to gather data related to cardiovascular activity.

The project will consist of several features:

- code to turn on the camera flash and record red pixel values coming from the image sensor
- code to extract meaningful information such as heart rate from the incoming red values
- a user interface to relay the extracted information

Background

Photoplethysmography (PPG) data does not look like ECG data. In the first image below, you can see PPG (white) and ECG (black) data superimposed. Note that where there is an R peak in the ECG data, there is a trough in the PPG data. Also note that there are several peaks and troughs in the PPG data that look like heartbeats but aren't; be sure to account for these in your algorithm. Note that there is some delay in PPG signal compared to ECG signal because it takes some time for the blood to travel from the heart to the finger tip. The first image is a simplified version without indicating such delay time with the use of smartphone camera. The second image is PPG signal using a clinical device called pulse oximeter, and it shows the delay time between the ECG and PPG signals. As you can tell the PPG signal from smartphone camera is much noisier than the measurement from pulse oximeter. One of the challenge is to figure out the periodic heartbeat pattern in the signal and extract useful information from such noisy data. According to our observation, smoothing and butterworth filters smooth out peaks of both noise and useful PPG peaks/trough.



[2]

Figure 1. PPG and ECG overlayed using smartphone camera (without indication of delay between PPG and ECG)

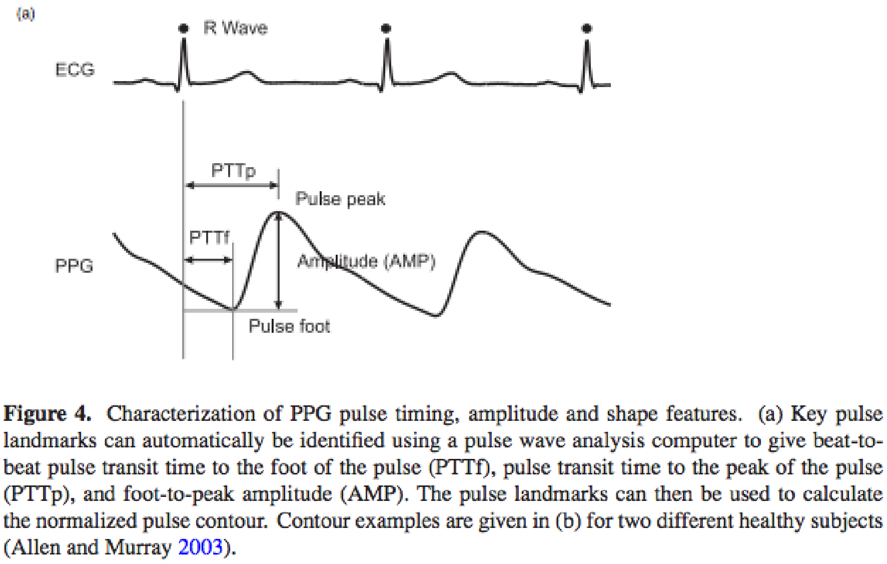


Figure 4. Characterization of PPG pulse timing, amplitude and shape features. (a) Key pulse landmarks can automatically be identified using a pulse wave analysis computer to give beat-to-beat pulse transit time to the foot of the pulse (PTTf), pulse transit time to the peak of the pulse (PTTp), and foot-to-peak amplitude (AMP). The pulse landmarks can then be used to calculate the normalized pulse contour. Contour examples are given in (b) for two different healthy subjects (Allen and Murray 2003).

[1]

Figure 2. PPG and ECG overlayed using pulse oximeter (Delay between PPG and ECG is shown.)

Assignment Description

In this assignment, we will be using the mean of red pixel value over all pixels in each image frame as our PPG signal.

You will be responsible for **storing up the incoming values** (like the windows or buffers we've been using throughout the semester), **extracting information from the buffered data**, and **making this information available to the user** in real time through the UI.

Requirements:

- Write code to obtain a fixed number of red pixel value every second. (Currently, the provided code obtains roughly about 3-7 average red pixel values (one pixel value for each frame) every second.)
- Write code to build up a buffer (array) of incoming values. The size of buffer is up to you -- experiment to find an effective size.
- Write an algorithm to capture individual beats from the data, and use this information to calculate a heart rate in beats per minute (BPM).
- Relay the BPM-related information to the UI. You can build a UI from scratch (using previous projects as a reference if that's helpful) or you can modify one of our previous assignments -- like the step-detection or client visualization projects -- to work as a photoplethysmography app.
- Extra credit: see below for extra credit options. (Please note that the extra credit options are not alternatives to the above requirements, only additional features that you can integrate into the app for additional points.)

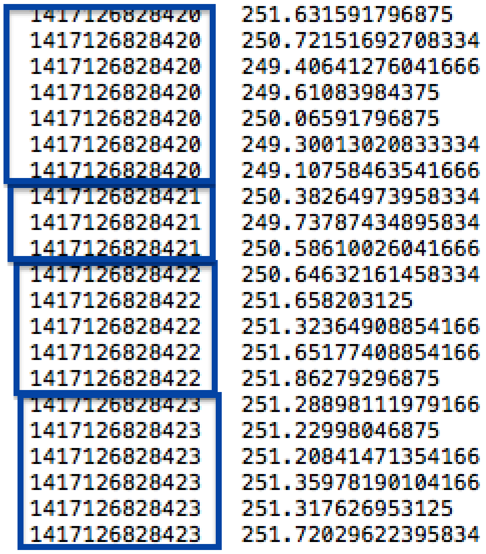
Code

Download this Android Project. Most of the code for capturing data from the image sensor is provided in this project. CameraPreview class contains a wrapper class called Preview. The onPreviewFrame() method in Preview is where you will be adding most of the code. Currently, the code that we provided does the following:

- set the resolution to be the lowest possible: 128x96 (mPreviewSize.width=128, mPreviewSize.height=96)
- store the red pixel values in a 1d array for each frame.
- calculate the mean of red value for the middle 400 pixels (nCenterPixels in the code) in 1d array. (Feel free to change nCenterPixels and compute the mean for bottom half of the image array since the bottom part of the camera is closer to the flash and better-illuminated.)
- write the red pixel values and time stamp in a text file in Downloads/MyAndroid folder in your phone.

Understanding PPG signal

- First, try to collect some data using the code provided here. **The current code collects data once the app is launched, so you have to put your finger against the camera before you start the app.** You will have to later modify the code to turn on the flash and start collecting data only when "start" button is pressed. The red pixel values and time stamp will be stored as a text file in Downloads/MyAndroid folder in your phone.
- Visualize the data using excel or other tools and try to find the troughs in PPG data that corresponds to ECG peaks. If you can't tell with your eyes, there is little chance that your algorithm could work.
    - **Note**: Refer to speech detection assignment to see how you accessed your audio data that is stored on your phone. For mac user, please download Android File Transfer app to transfer files from the phone to the computer. You don't need this app if you use window computer. File Explorer app might be useful to access to files in Downloads folder right on the phone without connect via USB.
- You can wear the chestband while collecting PPG data to obtain the ground truth value of ECG signal. You can use your pulse rate from chest band and the timestamp for PPG signal to calculate how many heartbeats are roughly present within some number of PPG samples. This might be helpful in finding the troughs in PPG that corresponds to heartbeat.
- Read through the code especially mPreviewFrame() method in Preview class in CameraPreview.java, and understand roughly how the average value of red pixels is calculated for each image frame.
- Use timestamp to obtain a constant number of red pixel value every second. Although Galaxy Nexus camera has frame rate of 24 frames per second (fps), the maximum fps is not achieved in our code. The number of frames recorded per second also varies in the range of 3-7 frames as you can see in the following snippet of red pixel. Blue box in the figure corresponds to one second window. Since timing is important for PPG signal, you have to ensure to store a fixed number of red pixel values every second in buffer. This means that for example, if you choose to store 2 red pixel values per second, you have to split each blue box window into half and take the average over each half to get 2 PPG values per second.

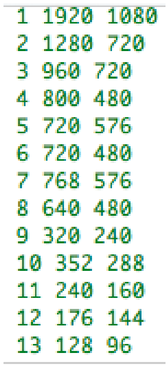


Index	Timestamp (second)	Red value
65	1417126828420	251.631591796875
66	1417126828420	250.72151692708334
67	1417126828420	249.40641276041666
68	1417126828420	249.61083984375
69	1417126828420	250.06591796875
70	1417126828420	249.30013020833334
71	1417126828420	249.10758463541666
72	1417126828421	250.38264973958334
73	1417126828421	249.73787434895834
74	1417126828421	250.58610026041666
75	1417126828422	250.64632161458334
76	1417126828422	251.658203125
77	1417126828422	251.32364908854166
78	1417126828422	251.65177408854166
79	1417126828422	251.86279296875
80	1417126828423	251.28898111979166
81	1417126828423	251.22998046875
82	1417126828423	251.20841471354166
83	1417126828423	251.35978190104166
84	1417126828423	251.317626953125
85	1417126828423	251.72029622395834

- Currently, for every image frame that is recorded, the red value for 12288 (128x96) pixels are stored in an array. Try different resolutions and plot the data in excel to see which resolution gives you the best option. Choose the resolution that works the best. You can change resolution in OnMeasure method in CameraPreview.java by changing the index in the following line:

  mPreviewSize = mSupportedPreviewSizes.get(index);

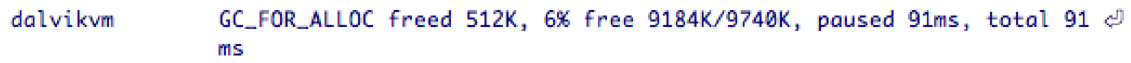
  *Index for different resolutions supported by Galaxy Nexus camera is given below.

Index	Resolution	
1	1920	1080
2	1280	720
3	960	720
4	800	480
5	720	576
6	720	480
7	768	576
8	640	480
9	320	240
10	352	288
11	240	160
12	176	144
13	128	96

  - Note: You will find that better resolution gives you more significant fluctuation in red values. But if you choose the best resolution possible (1920x1080), the app might skip a lot of frames and get super-slow, because we are doing color conversion and finding average over several pixel values (2073600 values) every half-a-second or so. One way is to take the average of the red pixel values in the center area of the image frame. Or, you can try lower resolution. Feel free to improve the efficiency of the current data collection process.

  - You might get the following warning in LogCat. `GC_FOR_ALLOC` means there were not enough free memory to fulfill an allocation request, so a garbage collection was necessary. But, the data collection should still works. We have not come up with the best way to resolve this. We will update on piazza as soon as we find a way to avoid this error. If you find a good solution for this, please post on piazza.



```
dalvikvm        GC_FOR_ALLOC freed 512K, 6% free 9184K/9740K, paused 91ms, total 91 ⏎
                ms
```

- Write the algorithm to detect troughs in a window.
- Calculate heart rate in beats per minute by either:
    - finding the distance between troughs and converting it to bpm. Using chest band measurement, find the conversion factor to convert from the distance between consecutive troughs to beats per minute. For example, if the distance between two troughs is 2 and chestband reading is 70 bpm, your conversion factor is: 70/ 2= 35. Therefore, for the distance= 2.5, your bmp is 2.5*35.
    - counting number of troughs within a certain window length and converting it to bpm.

  Note: print out the pulse rate in LogCat for now during debugging process . Once this works, you will modify the UI to return the pulse rate on the screen.

- Collect data for different scenarios such as stationary and exercise and see if there is difference in PPG signal and bpm.
- Modify the code to turn on the flash and start collecting data only when “start” button is pressed.
- Return the pulse rate on the UI screen. Refer to previous assignments to see how you did it.

Notes:

- If the signal is inperiodic, check the following:
    - Make sure to place your finger on the back camera before the app is launched. Data collection starts once the app is lauched.
    - Make sure to block the entire camera lens with your finger. Make sure camera flash is on during data collection.
    - Try not to move your finger on the camera while collecting data. Remember even slight movement could mess up your data. When you are collecting data for exercise, try to do exercise that minimize the movement of your finger on the camera.
    - You don’t need to press your finger against the camera lens; this might cause some redness in your finger and affect PPG signal.
    - Try breathing heavily while collecting data if the data do not fluctuate much. This will increase amount of blood pumped out of the heart; therefore, this increases the amplitude of the PPG signal and makes it easier to differentiate small jitters from noise and actual PPG peaks and troughs.
    - Try different ambient lighting conditions: dark and bright.

- Remember the measurement of your app doesn’t need to be as accurate as clinical devices. Since the PPG data is super-noisy, the measurement won’t be that reliable. As long as your app shows a reasonable heart rate for baseline and some changes in heart rate during exercise or other activities that cause elevated heart rate, you will receive credit.
- To get an idea about how a PPG app looks, you can try some heart rate monitor apps on the Google Play store.
    - Face: What’s my heart rate

        - Finger: Instant Heart Rate, and Heart Beat Rate.

Extra Credit

a. Visualization of PPG signal

Show time-series graph of PPG signal in real time. Refer to client visualization assignment.

b. Measuring Cardiac Output per Time

An average user might not know what heart rate tells about his or her health. If you want to give more informative output on the UI screen for the user, you can try one or more of the following:

1. Healthy heart rate:
    a. You can research the healthy heart rate for different age range. Your app can:
        i. let the user enter his or her age in a textbox. Healthy heart rate could vary depending on your age.
        ii. let the user select whether the user is resting or exercising, and whether the user consumes caffeine within the last few minutes.
        iii. according to the user's input, determine whether the heart rate is within the healthy range of the appropriate age group, and show the result on the screen.
    b. You might not be able to find healthy heart rate for caffeine intake and exercising online. You could measure these values on your own. Assuming that your heart rate is healthy, you can collect your heart rate (and/or stroke volume) data for different scenarios: caffeine intake, stress and physical exercise in advance. Feel free to use chest band for better accuracy. Since this data could vary for different individuals, all of your teammates could measure heart rate to get a range of healthy heart rate for each scenario. Now, you can determine if a given heart rate is healthy by checking if the value is close to each of those range within some standard deviation. This is just one method. Feel free to come up with other methods.

Note: Remember there could be many confounding factors in this case. Fast heart rate could be caused many factors such as physical exercise, stress or caffeine intakes. You cannot conclude that the person is not healthy just by looking at the fast heart rate. He could be exercising, or maybe he is nervous. So, you couldn't conclude whether the heart rate is healthy without knowing additional information such as whether the person is stationary or not, or if the person has just consumed caffeine.

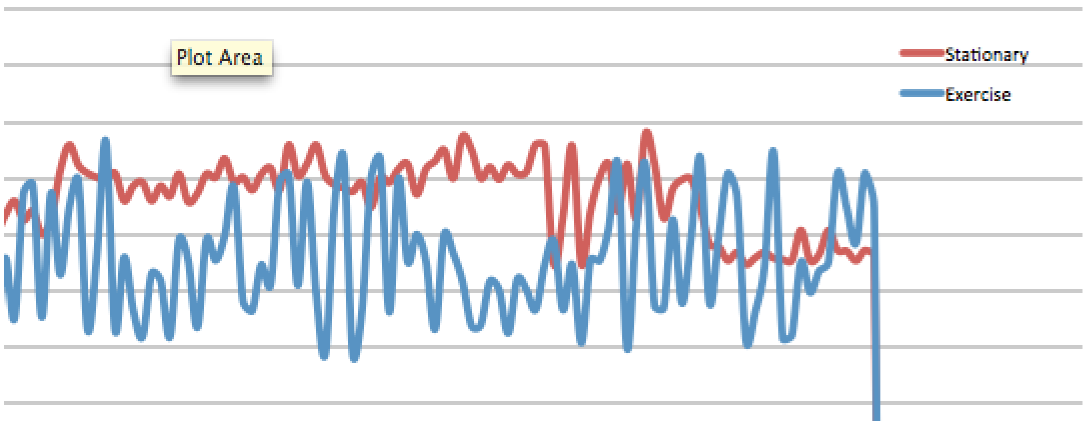
2. Life expectancy
    a. You can use heart rate to calculate the estimate of life expectancy of the user. Calculator is here.

3. Different levels of blood stroke volume & cardiac output

Now that you have completed measuring the heart rate, you could measure other features from the PPG signal besides pulse rate.

Stroke Volume

One of the features that you can easily measure is stroke volume, which is the amount of blood that the heart pumps through the circulatory system in a minute. The amplitude in PPG signal is proportional to the stroke volume. Although you can't compute exactly how much liters of blood is pumped out, you can see the difference in the amount of blood for steady position vs. during exercise as shown in figure below. Standard deviation might be a useful variable to indicate stroke volume.



Cardiac Output

The cardiac output is another indicator for stress, physical movement or other physiological characteristics of the user. Cardiac output can be computed as below:

Cardiac output = stroke volume × heart rate [3]

If you are exercising, your muscles needs more energy (more oxygen); therefore, your heart must supply more blood to provide more oxygen to the muscles. The heart could supply more blood either by beating faster (higher heart rate, shorter distance between troughs) or by supplying more volume of blood (higher stroke volume). See if both effects are present in your PPG signal during exercise.

By thresholding, you can classify cardiac output into two basic levels: baseline vs. elevated cardiac output. Keep in mind that threshold value will be different for different person.

Deadline

The assignment is technically due **12/5/2014** (Friday) **at 1:30pm** (at the beginning of the lab). But, there will be no penalty if you submit it at Friday midnight. Please demo the app to us during the lab, since that’s the last day of class, and we won’t hold any TA hour after Dec 5. If you plan to demo after Friday, please make an appointment with one of us.

Please submit the following in a zip file on moodle:

1. Your Android project
2. A short write-up that includes
   1. your teammate names
   2. how each teammate contributes to the assignment
   3. experiments that you did and interesting observations
   4. explanation for the extra-credit section (if you did)
   5. accuracy of your PPG app.

      Accuracy= 100% x |ground truth - PPG measurement| / ground truth

Citation

1. http://iopscience.iop.org/0967-3334/28/3/R01/pdf/0967-3334_28_3_R01.pdf
2. http://www.gizmag.com/pulse-phone-heart-rate-app/16962/
3. http://www.webmd.com/heart-disease/tc/cardiac-output-topic-overview
4. http://www.medicinenet.com/script/main/art.asp?articlekey=7524