Remote Monitoring of Heart Rate using Multispectral Imaging

Group 2, 18-551, Spring 2015

Michael Kellman
Carnegie Mellon University
mkellman@andrew.cmu.edu

Bryan Phipps
Carnegie Mellon University
bgp@andrew.cmu.edu

Sophia Zikanova
Carnegie Mellon University
szikanov@andrew.cmu.edu

Abstract:

There are a variety of fields working toward remotely extracting physiological indicators. We have implemented a method to remotely measure heart rate and other physiological indicators via Photoplethysmography (PPG) using an Android camera. We anticipated that key issues would be from motion in the subject, as well as motion with the camera, and illumination changes on the region of the interest. Our goal was to improve on prior implementations. To evaluate our implementation we compared our estimate of heart rate to independent measurement of heart rate using a chest strap; we assumed the chest strap represented ground truth and used this to determine our measurement errors.

Introduction to optically measuring heart rate:

The current, most accurate, method for extracting heart rate from a subject is via an electrocardiogram (ECG). ECG's are measured using several leads on the body surface around the chest and sense the electrical impulses representing the electrical activation of the heart. The signal represents the process of the cells in the heart depolarizing and then repolarizing causing the to heart to contract [10]. Post processing on these signals results in a one-dimensional transient signal. One pulse on an ECG reading can be decomposed into PQRST complexes; each represents different stages of the heart polarizing/de-polarizing.

The area of research encompassing optically extracting of the heart rate is very well characterized and developed. There are two prominent styles of measuring heart rate optically. The systems are differentiated by how the light passes from the emitter to the detector, either the light enters the tissue and is reflected into the detector or the light is transmitted through the tissue into the detector. Both methods utilize the fact that the hemoglobin in the blood has specific absorption properties. The relationship between absorption and wavelength of light is very well characterized as seen in Figure 1.

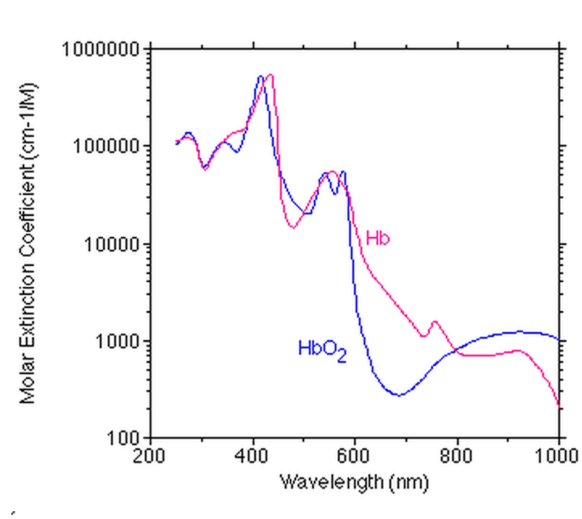


Figure 1: Total hemoglobin is illustrated in red and oxygenated hemoglobin is illustrated in blue. Note the local maximum around 510 nm characteristic of green light [8].

Motivation:

There are many benefits to being able to remotely measure and record heart rate. The benefits are medical, computational, and psychological. Being able to measure your heart rate is one way that people are able to track their health and fitness activity. Remote heart rate measurement could provide a method that would lead to improved accuracy in monitoring anxiety, sleep patterns, and cognitive stress. This would be beneficial for psychological tests such as polygraph tests, as the subjects would not know they are being monitored, plausibly yielding a higher accuracy reading. Remote tracking would also allow for further development of emotion recognition systems.

This kind of monitoring would also have the benefits of not having to connect the subject to an ECG and would eliminate the cost of electrodes and the restriction on motion. This would be beneficial for the comfort of hospital patients. A more specific application would be monitoring babies at home, as it would be difficult and potentially dangerous to keep devices attached to them for extended periods of time. This would allow parents to confirm that their baby's heart rate is normal. As the population continues to age further, it would be helpful for physicians and statisticians to have their patients track factors such as heart rate more continuously. This would allow physicians to prescribe more effective treatments and statisticians to develop more significant trends.

Related works:

Remote monitoring of heart rate has been shown to be possible from a variety of methods, which include but are not limited to: laser doppler, microwave doppler, and thermal imaging [1]. Verkruysse et al., suggest an implementation of monitoring heart and respiration rate using ambient light and a web camera [6]. McDuff et al. suggest a similar method of using ambient light, but image with a multispectral camera as to gain a greater number of input sensor signals to separate [1].

Another notable method for extracting heart rate utilizes minute rigid head motions. Balakrishnan et al. exploits this physiological event to extract heart rate and heart rate variability (HRV) [7]. They extracted motion due to the influx of blood to the head region and then separated out heart rate by monitoring the transients of motion along the principal components of the head's trajectories [7].

Android and iOS devices have been used as a platform for many developers to gain quick and easy access to visual sensor information. ViTrox Technologies has developed an iOS app that is similar to the McDuff et al. implementation. Their iPhone app uses the front facing camera to monitor color fluctuations in the face and from these signals, they extract heart rate [15].

Methods:

At a functional view, the Android tablet device will record a video of a person's face and the remote heart rate measurement application will measure and display the estimated heart rate. The video will be taken from a Motorola Xoom tablet. The Xoom has a front facing camera with two-megapixel resolution. The video on the Xoom is 720p at 30 frames per second [11]. We researched using an infrared camera attachment to provide more spectral information for this project. The attachment details specifications for the Seek Thermal Infrared Camera. This IR camera plugs into the phone or tablet via USB. This attachment has a 32k pixel resolution and runs at 5-10 fps [12]. Unfortunately Seek did not release the API to us and as a result its functionality was dropped from the project.

The first step was to process existing pre-recorded videos with known heart rate measured at the same time. We developed the signal processing to measure the heart rate from the pre-recorded videos and compared our heart rate reading with the known value, giving us a percent error. The Polar T31 chest strap was used for continuous heart rate monitoring. Once validation of the signal-processing algorithm was complete and satisfactory results were reached, the next step was to implement the application on the Xoom tablet utilizing the live camera feed with stabilized motion control. To set up this motion control, the subject placed the tablet on the table and rested their chin on a stationary surface; this reduced signal corruption due to motion. The subject also wore a chest strap, so that the true heart rate value could be measured. Our results were synchronized with the chest strap reading, yielding a percent error. Once validation was complete at this step, the constraints of requiring constant ambient lighting and stable motion control were to be relaxed. These factors were removed one at a time in order for us to evaluate the effects of each factor individually.

Figure 2: Polar T31 chest strap heart rate monitor [21]

Databases:

A database of videos was used to begin the initial algorithm development. There are existing open source databases containing videos, EEGs, ECGs, stimulation logs, and other data. We used the video and ECG derived heart rate for initial development obtained from the HCI Tagging Database provided by the Intelligent Behaviour Understanding Group (iBUG) at the Imperial College in London, England [13][17]. This database was created to research ways to explicitly tag information in social media. The corpus of information includes a variety of camera positions of stationary subjects as well as synchronized EEG and ECG readings. We used the front facing RGB camera videos and the synchronized ECG information from the database.

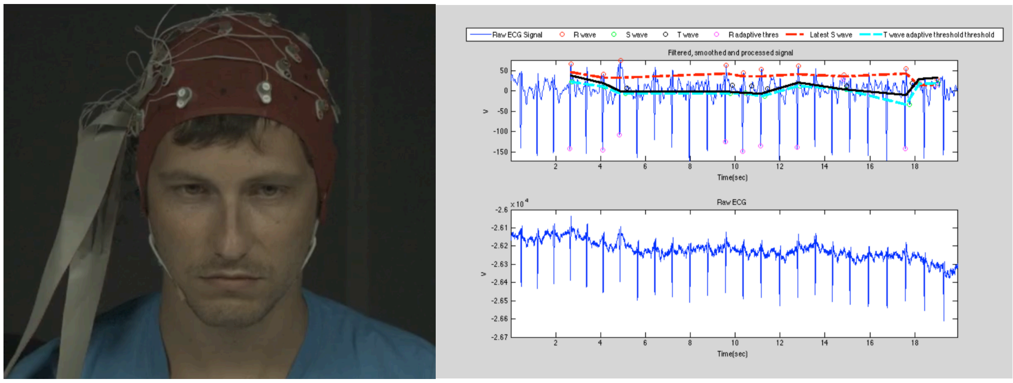
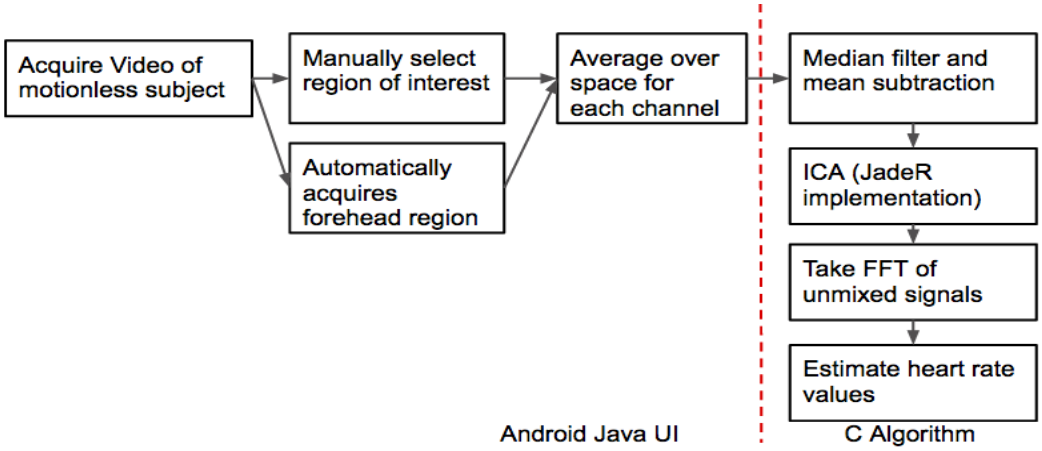


Figure 3 & 4: Screen shot of video from HCI database (left), Matlab figure of ECG values generated by QRS detection (right) [5][13][17].

Most of the algorithm validation involved gathering our own video rather than using this database. It was easier and faster to interface with our own video and recorded heart rate signals in Matlab than to use the database. Included in the Matlab code directory are a few videos and corresponding heart rate measurements.

Algorithms:

The signal processing flow is diagrammed in the block diagram below. The Android tablet implementation follows this signal processing flow. Left of the dashed line outlines the portion of the algorithm that is implemented in Java. Right of the dashed line outlines the functions that are coded in C++. In the following sections both algorithms are described, as well as the inter-process protocol for communication between them.


Acquire Video of motionless subject
Manually select region of interest
Automatically acquires forehead region
Average over space for each channel
Median filter and mean subtraction
ICA (JadeR implementation)
Take FFT of unmixed signals
Estimate heart rate values
Android Java UI
C Algorithm

Extracting the face and region of interest was done using the OpenCV Frontal Face Haar-Cascade Detection. This allowed the extraction of the face region and used it as input to the region of interest (ROI) algorithm. The ROI is centered with the intent to capture the upper forehead area to eliminate motion due to blinking. We find the location of both eyes based on the dimensions of the face box. The bounding box containing the left eye is used to calculate the bounding box containing the forehead. The equations governing how to calculate the bounding box containing the forehead was experimentally developed. Using this method to extract the bounding box containing the forehead automatically scales the ROI based on the size of the user's face, allowing the user's heart rate to be extracted at a varying distance. The dimensions and location of the forehead box are calculated once at the beginning of the acquisition due to computational limitations in finding the face with the Java implementation of the Haar-Cascade classifier.

The signal corresponding to each color wavelength was spatially averaged over the ROI to create a scalar time signal for each wavelength. In the case of our Android camera, there were three wavelengths (red, green, and blue). This three dimensional signal is made up of three transient signals that represent the average absorption fluctuations within the ROI. In this case, we are looking at the signals that are reflected off of the ROI. The reflected signal corresponds to the component of light that has not been absorbed by the subject.

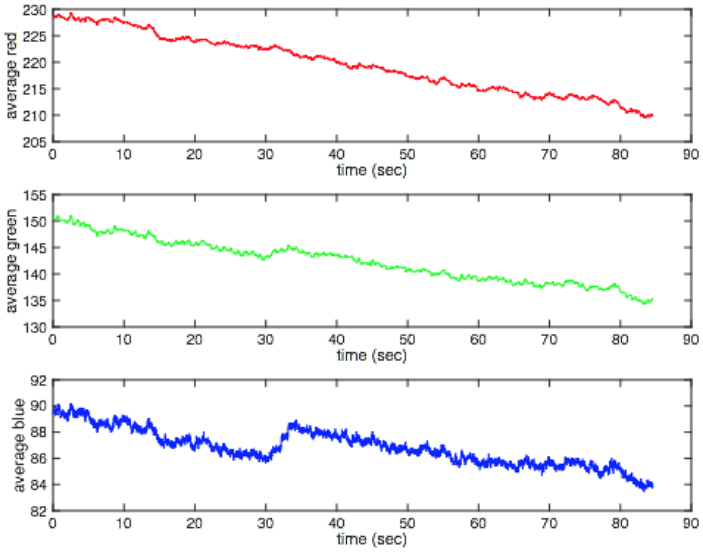


Figure 5: Spatially averaged color filter signals. The plots are color-coded by the corresponding color signal.

This three dimensional temporal signal are then fed from the Java side of the application into the C++ implemented algorithm. Below outlines the functionality of how the signals are separated.

The three-dimensional signal extracted from the previous steps encapsulates a mixture of unknown signals. The hidden signal of interest is correlated to blood volume pulse (BVP), which is heart rate. The other signals might represent other physiological signals, but they are not of interest and their meaning will not be hypothesized at this time.

First the mean is subtracted from these signals. If eyes are included within the region of interest, then the signals need to be median filtered to remove transient spikes. The transient spikes are a result of eye motion and blinking. This concludes the pre processing required before signal separation.

To separate our multidimensional transient signal, we used a method known as Independent Component Analysis (ICA) [1][2]. The goal of ICA is to separate a mixture of signals into statistically independent signals [3]. There are a handful of methods to accomplish this goal. Before deciding which particular algorithm is best for our application, we needed to understand the fundamental differences between them and from where they are initially derived. There are two traditional interpretations of statistical independence in terms of ICA, taking two main definitions: Minimizing Mutual Information (MMI) and Maximizing Non-Gaussianity [3].

In McDuff et al., their implementation used algorithms based around Maximizing Non-Gaussianity of their signal [1]. McDuff et al. used the JadeR implementation of ICA with a five dimensional input (other dimensions representing different wavelengths). Our implementation will work with the three sensor signals and result in three separated source signals. We use the real valued JadeR implementation of ICA developed by Jean-Francois Cardoso [19]. We then view the unmixed signal in the frequency domain and estimate the heart rate. The measurement involves finding the maximum peaks in the frequency domain between 45 and 200 bpm for each unmixed signal. The greatest peak between 45 and 200 bpm of all three signals is our estimate of the energy of the heart rate signal. The frequency (in bpm) associated with that peak is our estimate of the heart rate. That value is returned to the Java side of the code and is displayed to the user on the right side of the screen.

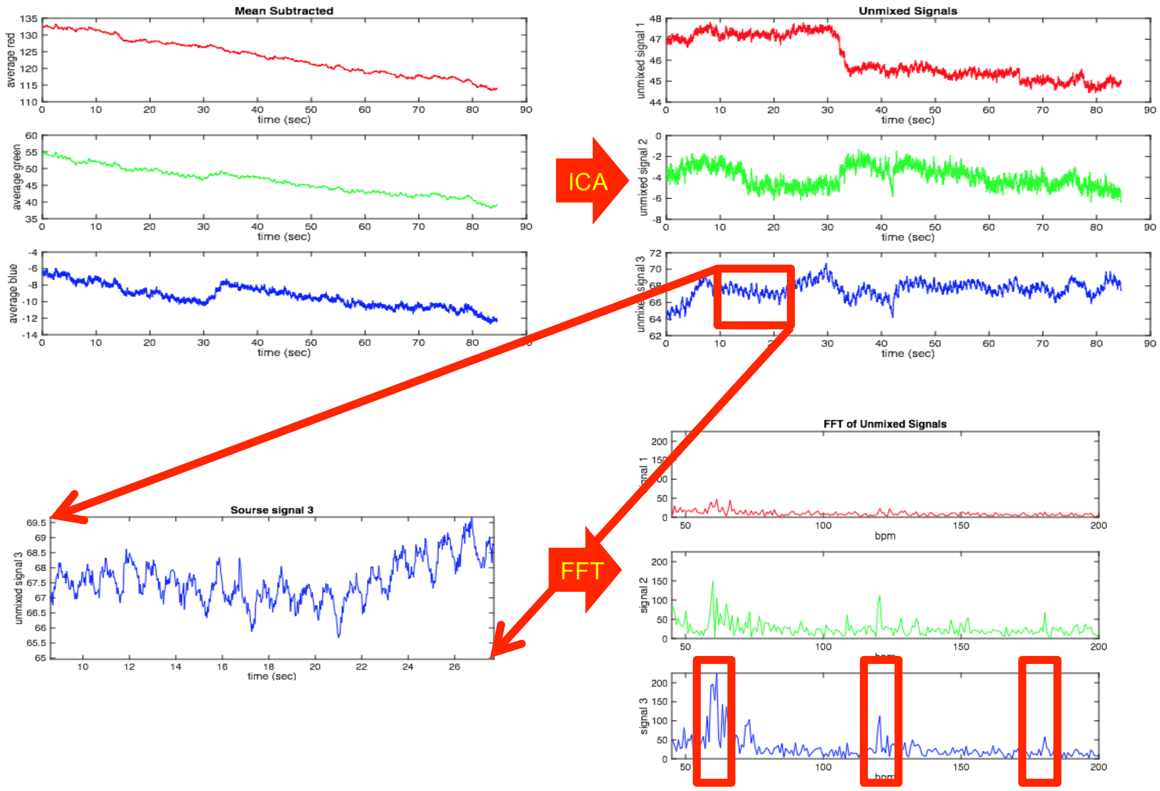


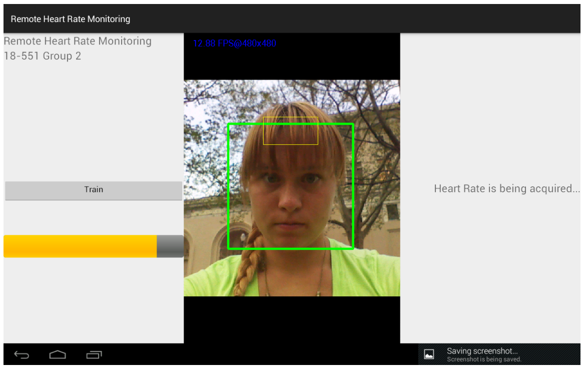
Figure 6: Above is a flow chart of the stages that are applied to the signals. These signals were extracted from a video recorded from the Android tablet.

Android Implementation:

The specific on-device implementation is as follows. With the first camera frame the face is found using the OpenCV haarcascade_frontalface_default classifier. Then, the forehead box is calculated and reused for every subsequent frame of the acquisition. Each camera frame, red, green, and blue values are extracted and spatially averaged, resulting in three floating-point values. Each float value for the red, green, and blue values are appended to an ArrayList. Once we have sufficient data, 250 discrete time points in the three ArrayLists, we begin to process the data. We use a custom class, rgbSig, to hold the data we pass between Java and the Java Native Interface (JNI). The class holds three float arrays, one for each signal, an integer to hold the frame rate of the acquisition, and four floats to hold the resulting peaks that are our estimates of heart rate, as well as our best estimate of heart rate. On the 250th camera frame the red, green, and blue ArrayLists are copied into the float arrays of the rgbSig class. The data is then passed as the argument of the JNI function that executes the algorithm that is implemented in C++. The C++ function handles the wrapping and unwrapping of the Java rgbSig class and passes the correct three float arrays and frame rate arguments to the main C++ function that extracts the heart rate as detailed in the algorithms above. The three estimates and the one best-estimated heart rate values are returned. The best estimate is shown to the user on the right side of the screen. The red, green, and blue ArrayLists are then released and reinitialized for the next heart rate acquisition instance.

Demo:

This is the final user experience that was demonstrated during the capstone exposition. The heart rates that were calculated on the tablet were compared to a finger pulse oximeter attached to the subject to highlight the accuracy of our Android application. During the exposition we used a poster board to provide a white backdrop to ensure constant contrast between the subject and the background.



Remote Heart Rate Monitoring
18-551 Group 2

12.88 FPS@480x480

Heart Rate is being acquired...

Saving screenshot...
Screenshot is being saved.

Figure 7: Android application demo splash screen

Results:

The data points presented in Figure 8 represent remote heart rate measurements versus the ground truth measurements obtained via the chest strap. The blue line representing y = x lends a qualitative evaluation of the accuracy of our algorithm.

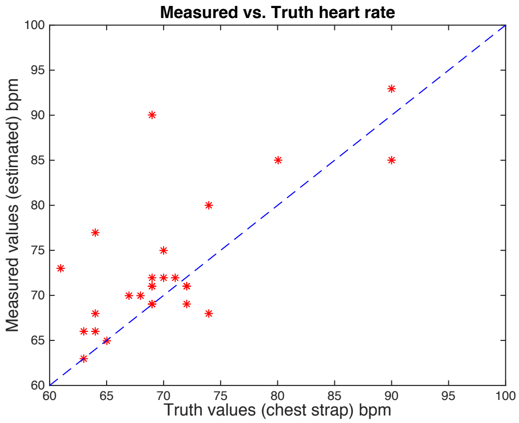


Figure 8: Measured heart rate versus Truth heart rate

Figure 9 is a Bland-Altman plot. The purpose of the Bland-Altman plot is to quantitatively evaluate the bias and spread between these two techniques for extracting heart rate. The figure is a plot of the absolute difference between measurements versus the average of the two measurements. The absolute bias from our testing data is roughly <3 bpm.

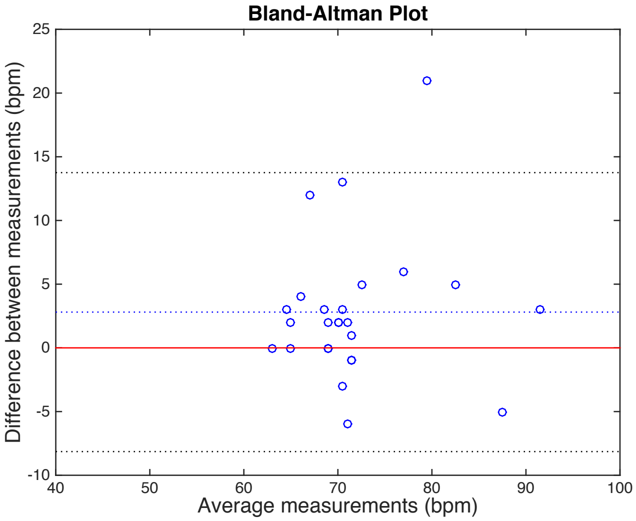


Figure 9: Bland-Altman Plot

Most finger pulse oximetry devices and fitness monitors have an error of roughly +/- 3 bpm. The chest strap claims to have +/- 1 bpm for constant heart rate, however the actual heart rate for healthy subjects is not constant [20]. The average absolute error between our estimates and the truth measurements was approximately 4 bpm.

Limitations:

We faced multiple limitations due to the technical specifications of the Android device. The tablet we used, had a camera with a frame rate of 30 fps, which was further reduced to 15 fps due to the processor's competition with the signal-processing algorithm. We researched locating the face every frame as a method of tracking the face but this brought our frame rate down to roughly 3 fps. Three frames per second is not a sufficient temporal resolution to extract heart rate and would ultimately take a very long time to acquire the necessary amount of frames. Given a device with more processing power and a camera with a higher frame rate we could acquire data with greater temporal resolution.

The best case for performance is achieved when the user and camera are motionless and the user's heart rate is constant. When the user's heart rate is constant, all of the energy in the heart rate signal is at one frequency value; this produces a frequency response with a discriminative spectral peak. If the user's heart rate is highly variable, then information in the frequency response will have degraded spectral resolution. As a consequence, there will be no clean single peak that coincides with the user's heart rate. The solution to this problem is accessible with a camera that has a higher frame rate. This would give us higher temporal resolution and enough samples to analyze the short-time spectral information in the signal, allowing us to capture changes in a varying heart rate signal.

Future Works:

Given an extended period of time to work on this project, we would put our efforts into implementing a computer vision algorithm to stabilize the video signal in order to reduce the effects of motion. We would approach this by researching two algorithms, which are outlined below:

Using the face as a template, we could implement Lucas-Kanade alignment to calculate the warp that would transform the current frame to the template frame. This would involve solving a gradient descent problem to estimate the warp that would minimize the error between the template and the current image. Each iteration of the gradient descent algorithm, the warp would be updated by a differential warp factor.

The second proposed solution is similar, but instead of minimizing the error between a warped image and a template, we can identify landmarks on the face in each frame and warp the image with the objective of lining up the landmarks.

Division of Work:

Sophia:

- Android application development
- Designing user interface and application structure
- Using openCV Java functions to do face detection and image data parsing

Bryan:

- Overall algorithm integration with Android application
- Developed android data acquisition and packaging for JNI interface integration

Michael:

- Database research and testing
- Researching and testing ICA algorithm implementation
- Development and testing of Matlab Algorithm to extract heart rate
- Development and testing of C++ Algorithm to extract heart rate

References:

[1] D. McDuff, S. Gontarek, and R.W. Picard, "Improvements in remote cardio-pulmonary measurement using a five band digital camera," IEEE Trans Biomed Eng., vol. PP, issue 99, 2014.
[2] Li, Xiaobai, et al. "Remote Heart Rate Measurement from Face Videos under Realistic Situations." *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.
[3] http://en.wikipedia.org/wiki/Independent_component_analysis
[4] Brian Moore, http://www.mathworks.com/matlabcentral/fileexchange/38300-pca-and-ica-package
[5] Hooman Sedghamiz, http://www.mathworks.com/matlabcentral/fileexchange/45404-ecg-q-r-s-wave-online-detector
[6] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, “Remote plethysmographic imaging using ambient light,” Opt. Expr., vol. 16, pp. 21434– 21445, Dec. 2008.
[6] M.-Z. Poh, D. J. McDuff, and R. W. Picard, “Advancements in noncontact, multiparameter physiological measurements using a webcam,” Biomedical Engineering, IEEE Transactions on, vol. 58, no. 1, pp. 7–11, 2011.
[7] G. Balakrishnan, F. Durand, and J. Guttag, “Detecting pulse from head motions in video,” in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013, pp. 3430–3437.
[8] Hemoglobin Absorption Image, http://omlc.org/spectra/hemoglobin/
[9] M. P. Tarvainen, P. O. Ranta-aho, and P. A. Karjalainen, “An advanced detrending method with application to hrv analysis,” Biomedical Engineering, IEEE Transactions on, vol. 49, no. 2, pp. 172–175, 2002.
[10] ECG overview, http://www.emergsource.com/?page_id=90
[11] Xoom specifications, http://www.droid-life.com/motorola-xoom-specs/
[12] Thermal camera, http://obtain.thermal.com/product-p/uw-aaa.htm
[13] Database, http://mahnob-db.eu/hci-tagging/
[14] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic, “Local evidence aggregation for regression-based facial point detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 5, pp. 1149–1163, 2013.
[15] Vitrox Technology Implementation, https://appsto.re/us/kyjOE.i
[16] Finger Pulse Oximeter, http://www.amazon.com/gp/product/B004BJT9OE/ref=pd_lpo_sbs_dp_ss_1?pf_rd_p=1944687562&pf_rd_s=lpo-top-stripe-

1&pf_rd_t=201&pf_rd_i=B00HISWGZY&pf_rd_m=ATVPDKIKX0DER&pf_rd_r=0TNWFCZ5Q43
A0422CH3C

[17] M. Soleymani, J. Lichtenauer, T. Pun, M. Pantic., "A multimodal database for affect recognition and implicit tagging", *IEEE Transactions on Affective Computing*. 3: pp. 42 - 55, *Issue 1*. April 2012.

[18] http://sourceforge.net/projects/kissfft/

[19] Jean-Francois Cardoso, http://perso.telecom-paristech.fr/~cardoso/guidesepsou.html

[20] Polar T31 specifications,
http://www.polar.com/support_files/za/422573EF0043154142257792004F2532/17925349
_AXN300_ENG_B.pdf

[21] Polar T31 Image,
https://encryptedtbn0.gstatic.com/shopping?q=tbn:ANd9GcR5K95FtRNmAYqy9XuYUIpm9xd
7ND6e6APPwT6MPL2_321pOFk_IYMDFbCtyxNS2xQBdKWJc6sZ&usqp=CAY