MIDITHA
An Application for Non-Intrusive Heart Monitoring

A Project Report

submitted by

B S Pradyumna (12EC24)

Dedhia Yaamika Pravin (12EC34)

Pavan C M (12EC70)

under the guidance of

Dr. Deepu Vijayasenan

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April 17, 2016

ABSTRACT

Recent advances in sensor technology and mobile computing are now enabling practical non-intrusive approaches to measure vital signs and other biological signals. Remote measurements of physiological signals can provide comfortable health assessment without the presence of any electrodes or devices on the body. Our goal is to extract cardiac pulse rate and blood pressure from colour video recordings of human face. Our method is based on Eulerian Video Magnification (EVM) framework, which takes a standard video sequence as input, and applies spatial decomposition, followed by temporal filtering to the frames. The resulting signal is then amplified to reveal hidden information. EVM framework is a generic algorithm developed to reveal minute changes happening in real world. In our case it is used to visualize the flow of blood as it fills the face and also to amplify and reveal small motions which cannot be observed by naked eye. EVM typically magnifies colour variations to visualize flow of blood. We propose to estimate heart rate from the colour magnified video by observing the variation of pixel values for the recorded time. For estimating blood pressure we propose to calculate pulse transit time (PTT) from video sequences. It has been observed that there exists strong correlation between blood pressure and PTT, as PTT is often used as an index to arterial stiffness, and hence can be employed for indirect measurement of blood pressure. Conventional techniques for measuring PTT are based on measuring the electrocardiogram (ECG) signal using leads attached to the chest and measuring the photoplethysmograph (PPG) signal from a finger. The proposed method is both non-intrusive and non-contact based estimation, hence can be employed for continuous, cuffless and non-invasive measurement of blood pressure, which is more desirable for people who need regular monitoring of their blood pressure. We also implement the above method on a smartphone platform using the EVM framework which can run in realtime without necessitating the need of expensive hardware.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Problem definition

Regular and non-invasive assessments of cardiovascular function are important in surveillance for cardiovascular catastrophes and treatment therapies of chronic diseases. Resting heart rate, one of the simplest cardiovascular parameters, has been identified as an independent risk factor (comparable with smoking, dyslipidemia or hypertension) for cardiovascular disease [1]. Currently, the gold standard techniques for measurement of the cardiac pulse such as the electrocardiogram (ECG) require patients to wear adhesive gel patches or chest straps that can cause skin irritation and discomfort. Although non-contact methods may not be able to provide details concerning cardiac electrical conduction that ECG offers, these methods can now enable long-term monitoring of other physiological signals by acquiring them continuously in an unobtrusive and comfortable manner. Beyond that, such a technology would also minimize the amount of cabling and clutter associated with neonatal ICU monitoring, long-term epilepsy monitoring, trauma patient monitoring, and other cases where a continuous measure of physiological signals is important.

Thus, we have decided to take up this project, where we propose to estimate cardiac pulse rate of a person from the video stream of his or her face. We also plan to extend our implementation on a smart-phone platform based application such as an Android application.

1.2 Previous work

The human visual system has limited spatio-temporal sensitivity, but many signals that fall below this capacity can be informative. For example, human skin color varies slightly with blood circulation. This variation, while invisible to the naked eye, can be exploited to extract pulse rate [2], [3]. Similarly, motion with low spatial amplitude, while hard or impossible for humans to see, can be magnified to reveal interesting mechanical behavior [4].

Currently, proposed solutions for noncontact measurement of vital signs, such as heart rate (HR) and respiratory rate (RR), include laser Doppler [5], microwave Doppler radar [6] and thermal imaging [7], [8]. A common drawback of the aforementioned proposals is that the systems are expensive and require specialized hardware.

Photoplethysmography (PPG) is a low-cost and noninvasive means of sensing the cardiovascular blood volume pulse (BVP) through variations in transmitted or reflected light [9]. This electro-optic technique can provide valuable information about the cardiovascular system such as heart rate, arterial blood oxygen saturation, blood pressure, cardiac output and autonomic function [9]. Typically, PPG has always been implemented using dedicated light sources (e.g. red and/or infra-red wavelengths), but recent work [10] have used Independent Component Analysis (ICA) on human face video sequences to extract heart rate and Heart Rate Variability (HRV).

In this project we propose to use EVM based approach [11] to detect subtle variations appearing on the skin due to blood flow. The purpose of EVM is to overcome the limitation of human visual system and reveal more informative signals which are invisible by naked eye. Fig 1.1 shows the main framework of EVM.

1.2.1 Eulerian Video Magnification Framework

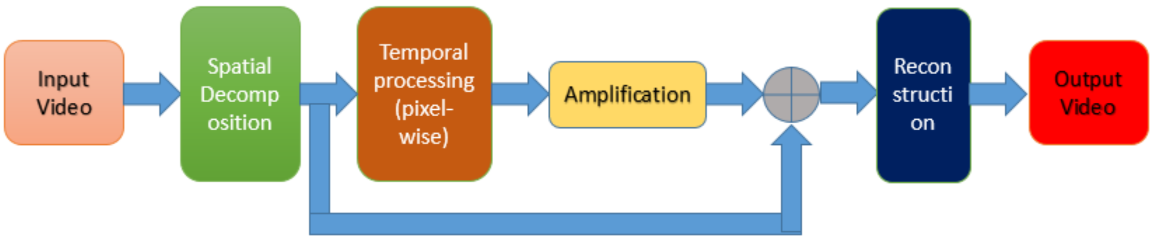


Figure 1.1: Block diagram of EVM Framework

The video is processed frame by frame and the subtle variations in the video are emphasized by the spatial and temporal processing, which involves several steps and can be described as follows:

- Decompose the standard video sequence into different spatial frequency bands and apply a full Gaussian pyramid. This mainly reduces image resolution, thereby reducing pixel count and hence lesser computational cost associated with temporal processing.
- Perform the temporal processing on the low resolution images by applying a band pass filter.
- Employ a magnification factor α to multiply the extracted band-pass signal.
- Reconstruct the signal by adding the magnified signal to the original one.

A detailed mathematical analysis of EVM framework is done in Overview section.

1.3 Motivation

The EVM framework enables us to target the barely seen changes in the pixels of a video and then amplify them. So what was previously a barely visible change in either motion or color is now made evident. By using this algorithm on face video, we will thus be able to 'see' a person's pulse by observing the color changes as blood is pumped to their face. Commercial pulse-oximetry sensors that attach to the fingertips or earlobes are inconvenient for patients and the spring-loaded clips can cause pain if worn over a long period of time. The option of monitoring a patient's physiological signals via a remote, non-contact means has promise for improving access to and enhancing the delivery of primary healthcare. The ability to extract vital signs from standard videos could be a breakthrough not only in medical diagnostics but could also be used by law enforcement agencies doing interrogations, psychologists during sessions, employers in an interview and even the casual aspiring mind reader. Presently, we do not find many smart-phone based applications for robust and accurate measurement of heart rate and blood pressure. However, by implementing EVM framework on a smart-phone, robust cardiac pulse monitoring will be made possible on a smart-phone platform. The above mentioned applications and the prospect of having a mobile application to monitor heart rate, motivates us to take up this project.

1.4 Overview

We have set two goals to be achieved in this project. First, it's the robust estimation of cardiac pulse rate. Second, is to derive blood pressure. For this we propose to use colour video recordings of a human face. Block level description of the EVM algorithm has been explained in the section 1.2. In the current section mathematical background, analysis of temporal filtering, post-processing applied on the colour magnified video to extract cardiac pulse rate and blood pressure has been explained.

Let $I(x,t)$ denote the image intensity at position x and time t. Since the image undergoes translational motion, we can express the observed intensities with respect to a displacement function $\delta(t)$, such that $I(x,t) = f(x + \delta(t))$ and $I(x,0) = f(x)$. The goal of motion magnification is to synthesize the signal

$$\hat{I}(x,t) = f(x + (1+\alpha)\delta(t)) \tag{1.1}$$

for some amplification factor α.
Assuming image can be approximated by first - order Taylor series expansion, the image at time t, $f(x + \delta(t))$ in a first-order Taylor expansion about x, as

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x} \tag{1.2}$$

Let $B(x,t)$ be the result of application of temporal bandpass filter to $I(x,t)$ and assuming displacement function $\delta(t)$ to be within the passband of temporal bandpass filter, then

$$B(x,t) = \delta(t)\frac{\partial f(x)}{\partial x} \tag{1.3}$$

In procedure of EVM bandpassed signal is amplified by α and added to the original signal

$$\tilde{I}(x,t) = I(x,t) + \alpha B(x,t) \tag{1.4}$$

Combining above equations, we get

$$\tilde{I}(x,t) \approx f(x) + (1+\alpha)\delta(t)\frac{\partial f(x)}{\partial x} \tag{1.5}$$

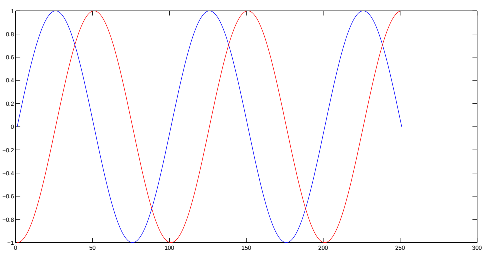
Assuming the first order Taylor expansion holds for amplified version of displacement function, $(1+\alpha)\delta(t)$, the processed output is

$$\tilde{I}(x,t) \approx f(x + (1+\alpha)\delta(t)) \tag{1.6}$$

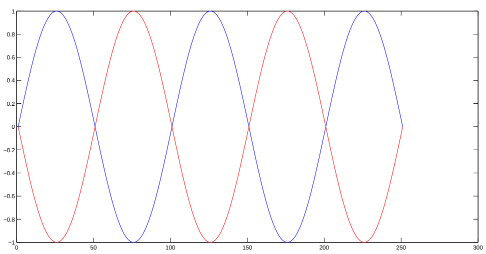
The above mathematical analysis has been dealt in detail in [11].

An optimal amplification factor applicable to all environmental conditions was chosen and video magnification was done with desired cutoff frequencies for the temporal bandpass filter. Once the desired video magnification was achieved, a face detection was done to remove the non-essential background parts. A particular pixel in the region of interest was tracked over time to construct a time domain signal. This time domain signal was used to estimate cardiac pulse by using autocorrelation. A peak is observed in the autocorrelation function at lags which are integer multiples of the period. This is illustrated for a simple sinusoidal signal in figure 1.2.

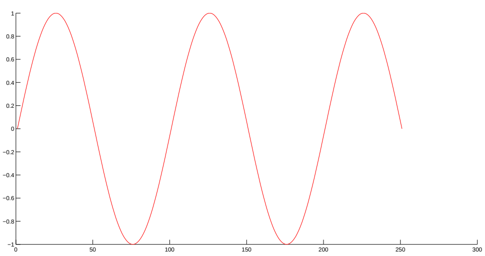
$$R(\tau) = E[X(t)X(t-\tau)] \tag{1.7}$$



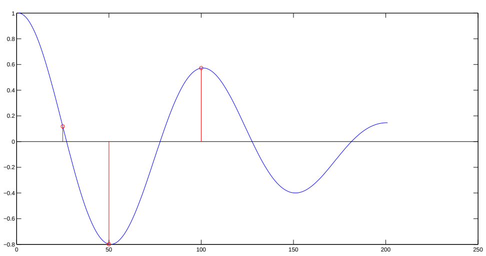
(a) lag = 25, autocorrelation = 0.1179



(b) lag = 50, autocorrelation = -0.7983



(c) lag = 100, autocorrelation = 0.5724



(d) autocorrelation function

Figure 1.2: Figures illustrating estimation of period from autocorrelation

In order to get a better estimate by considering intensity variations of pixels over the entire face, a collection of well illuminated 20 random points on the face were tracked. Information from all points were used by constructing a time series for every point on the face.

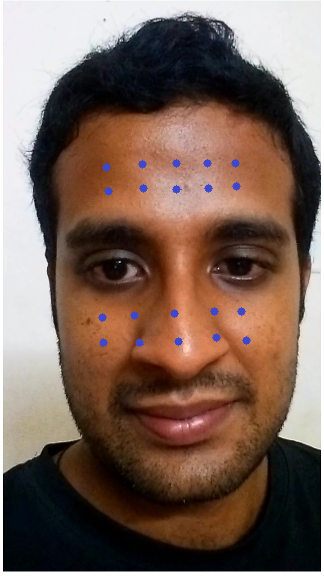


Figure 1.3: 20 points marked blue were tracked over time - Subject 6

Our Android Application - MIDITHA is based on the work done by Chambino [20] in his application - PULSE. However, we have implemented an improved algorithm that was developed to obtain accurate estimate of the Heart Rate. Our algorithm is based on Mode based period Estimation which requires the least computations for Android implementation. This algorithm is explained in the next chapter.

CHAPTER 2
Present Work : Estimation Algorithm

The MATLAB implementation of EVM done by [11] was open sourced and was available at [13]. The videos on which [11] have done analysis was also made available. The code provided by them was doing motion/colour magnification on the input video sequences. However the major drawback of the implementation was their choice of cut-off frequencies for temporal bandpass filter. Due to the prior knowledge about the range of heart rate of the subject, they had employed a very narrow passband ($\approx 0.6Hz$). As our aim is to estimate the heart rate, there is a need of a broader passband as there is no prior knowledge about the range of heart rate. This necessitated us to have a new dataset for detailed analysis.

2.1 Data Acquisition

For evaluation purposes data was acquired from 10 subjects consisting of 5 female and 5 male persons. Data consisted of 2 minute video recording of face, captured from front camera of a smartphone. Simultaneously a finger-tip pulse oximeter was placed on the finger of the subject to acquire ground truth values of the heart rate as shown in figure 2.1. These values were used for assessing the performance of our system and determine accuracy of our estimation.

Figure 2.1: Pulse oximeter was used for obtaining ground truth heart rate. Pulse oximter provides heart rate in Beats Per Minute (BPM) as well as oxygen saturation level

2.2 Estimation methodology

Video magnification was done for the captured datasets with optimal value for amplification factor. 20 random points which had good illumination in the video were selected for obtaining time domain signal by tracking the variations in pixel intensity values over time. The peak locations in these signals will generally correspond to the time instant at which heart beats. Therefore the period between peaks corresponds to the time required for 1 complete heart beat. This period will be employed to compute heart rate.

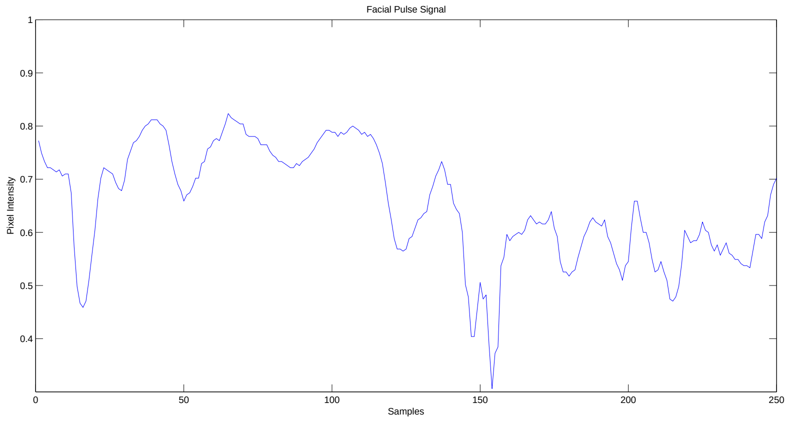


Figure 2.2: Plot showing variations of intensity of green component of pixel with time for subject 1

The challenge in estimating the period is that there is presence of other extraneous peaks which does not correspond to pulse rate, which appear in the time domain signal as shown in figure 2.2. The possible reasons for the presence of these artifacts is the interference of breathing rate with that of heart rate. Another possible explanation is the presence of aliased frequency components arising from the flicker of light sources in an artificially illuminated environment. A rectangular window frame of 8 seconds was constructed and period corresponding to this frame of signal was estimated using autocorrelation by determining peaks in autocorrelation function. In the next frame, another 8 second window with an overlap of 6 seconds with the previous window was considered. Heart rate estimate was given every 5 frames by analyzing the data of previous frames. Windowing operation is illustrated in figure 2.3

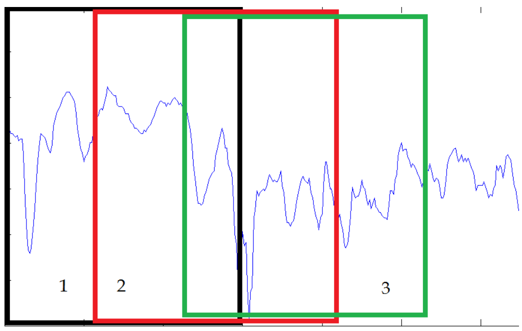
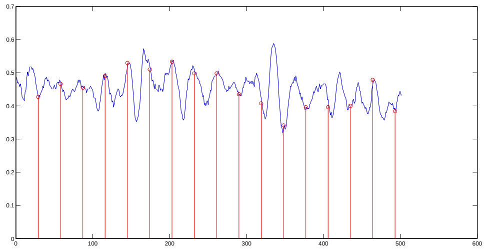


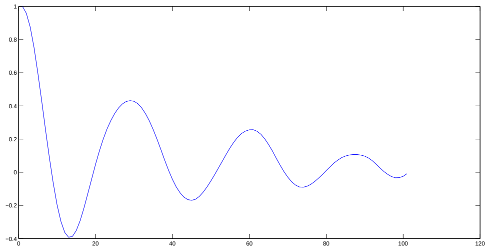
Figure 2.3: Figure illustrating windowing operation

The green channel was chosen for obtaining period from autocorrelation as it reportedly contains more information corresponding to heart activity when compared to red and blue channels [2].Figure illustrating the green component of the video sequence and it's corresponding autocorrelation for subject 1 is shown in figure 2.4

As evident from figure 2.4 the period estimation is not accurate due to the existence of erroneous peaks not representing cardiac pulse waveform. To overcome this drawback, multiple points on the face were tracked and information from all points were used for estimating precise period. The information from 20 points were analyzed using 3 different algorithms which is explained in the next section.



(a) Estimated period subject 1



(b) Autocorrelation function subject 1

Figure 2.4: Period estimation from autocorrelation

2.2.1 Mode based Estimation

Mathematically in a given vector, mode represents the element occurring at the highest frequency. Since the period detected in every frame obtained after windowing varies for different points on the face, the period occurring at the highest frequency is considered to best represent the heart rate of the corresponding time frame. The methodology is shown in block diagram figure 2.5.

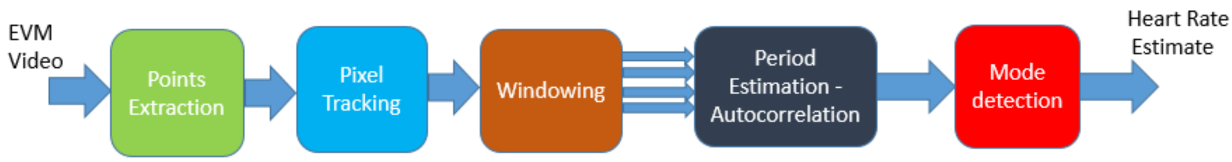


Figure 2.5: Block diagram - Mode based estimation

Heart rate estimate is given every 5 frames by taking mean of the estimates obtained every frame. Figure 2.6 depicts the histogram of 1^{st} frame of subject 1. It can be inferred that among 20 points on the face which were used for tracking, period of 29 (29 samples) is observed with highest frequency. Heart rate in Beats per Minute (BPM) is obtained by using

$$BPM_{estimate} = Fs * 60/T \tag{2.1}$$

Fs - Sampling Frequency; T - period estimate

The underlying assumption for the usage of above methodology is that majority of the points should capture information corresponding to heart rate and the number of points susceptible to artifacts should be minimal.

2.2.2 Modified Mode based Estimation - Weighted Average

Mode based period estimation used estimate obtained every frame to compute the heart rate at the end of every 5 frames.A modification to this method was applied by using all the estimates obtained for the period, by considering all 5 frames instead of a single frame. A histogram of all occurring periods is obtained for every 5 frames after windowing

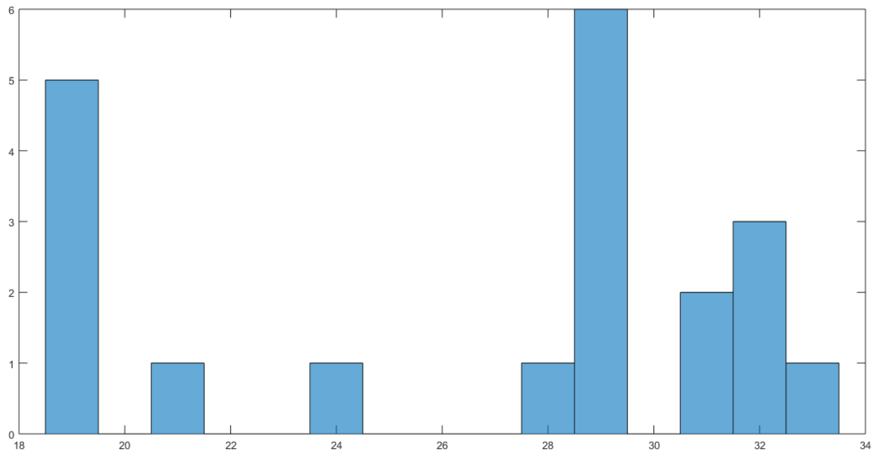


Figure 2.6: Histogram of periods for 1^{st} frame, subject 1

operation. A window of optimal length W is applied to the obtained histogram and summation of histogram values inside the window is obtained. The window is moved across the length of histogram with an overlapping length of $W-1$ between two successive windows. The window with the largest summation value is chosen as the window with the required period. A weighted average of the periods in this window correspond to the estimated period of the 5 frames of time domain signal used for period computation. Heart rate in BPM from estimated period is obtained using 2.1.

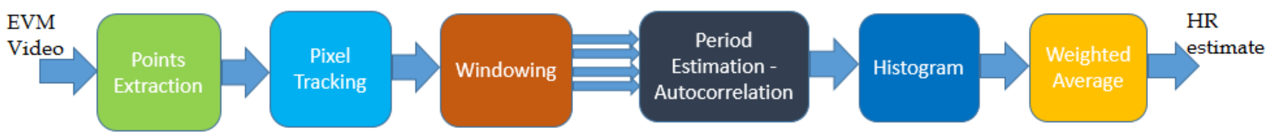


Figure 2.7: Block Diagram - Modified Mode based Estimation

As shown in figure 2.9, the highlighted bars correspond to the window with largest summation value when $W = 3$. A weighted average of the highlighted periods give the estimate of the corresponding period. This method mainly tries to eliminate periods obtained due to motion artifacts by merely considering more number of samples for analysis. A weighted average was employed so as to overcome the leakage effect - wherein the periods in and around the vicinity of the expected period were detected when large number of samples were considered for period estimation.

2.2.3 Period Tracking with Multiple Initializations

This method has been derived from the algorithms explained in [14] and [15]. This method involves 3 important steps for estimating the period of a given waveform. These are:

- Initialization

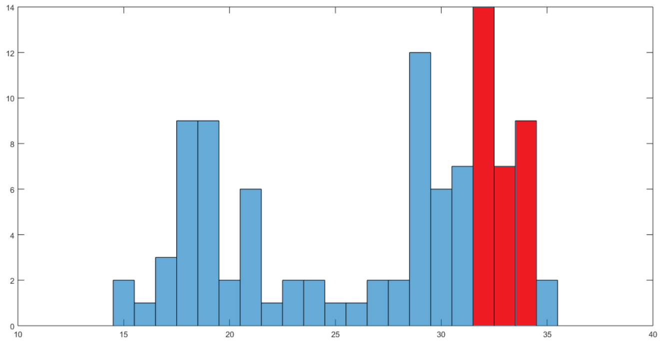


Figure 2.8: Histogram of periods obtained for first 5 frames - subject 1. Highlighted bars indicate the selected periods for $W = 3$

- Trajectory generation through forward-backward tracking
- Period estimation through Trajectory Strength maximization

Overview of the methodology

The video under consideration for Heart Rate estimation is colour magnified using the EVM framework. 20 well-illuminated points are then tracked on the subject's face in the colour magnified video. These pulse waveforms are then windowed as explained in figure 2.3. Corresponding autocorrelation waveforms are then used for peak - picking which is compiled in a matrix of size '20 x N_w' where N_w is the number of windows into which the pulse waveforms are divided. In order to find the period estimate of the k^{th} frame among the N_w frames, initialization (will be explained in the next sub-section) is carried out in every frame starting from frame 1 to frame k. Once initialization is done in a frame, forward-backward tracking is carried out to estimate k trajectories. Then, the strength S_k of all k trajectories are utilized and the period estimation corresponding to k^{th} frame is selected based on the trajectory having the maximum strength.

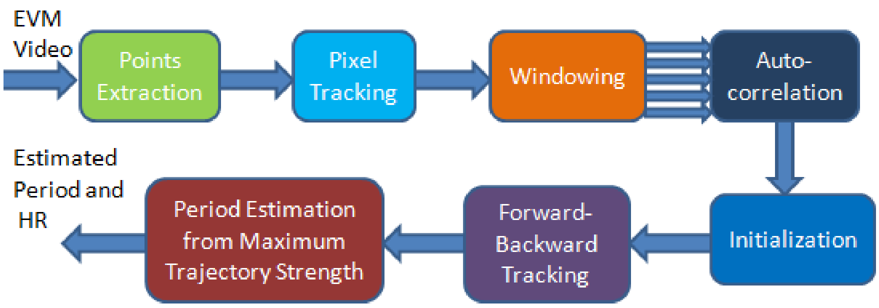


Figure 2.9: Overview of the Period tracking with multiple initializations

Initialization

We have 20 period estimates corresponding to each windowed frame from the 20 autocorrelations for the corresponding pulse waveform extracted from 20 selected points on the face. We find the mode of this 20 x 1 vector. If the frame to be initialized is the first frame, then the mode itself becomes the initialization for that frame. For the other frames, the mode of a given frame is checked if it lies within a threshold's distance from the period estimate of the previous frame. If the mode lies in a threshold's distance from the period estimate of the previous frame, then the mode itself becomes the initialization value, else mode is calculated for the 20 x 1 vector within a smaller window i.e. $[period_{estimate}(previous) - threshold\ ,\ period_{estimate}(previous) + threshold]$

Forward-Backward Tracking

In order to estimate the period of a given frame k, k initializations are carried out in every frame from 1 to k as explained in the previous sub-section. Once an initialization is done in a frame, a trajectory is constructed in both backward and forward directions. In backward tracking, the trajectory will constitute the mode value of the frames if the difference between the mode of that frame and the trajectory value of the next frame lie within a threshold. Otherwise, trajectory is filled with the trajectory value of the next frame. Similarly, in forward tracking the trajectory will either constitute the mode value of the frames or the trajectory value of the previous frame based on mode's distance from the threshold.

Period Estimation through Trajectory Strength Maximization

Once all k trajectories are constructed for the k^{th} frame, in order to choose the period estimation for the k^{th} frame we calculate the trajectory strength for all k trajectories. Trajectory strength is defined as the sum of the repetitions of the trajectory values in the corresponding frame vectors. The trajectory with the maximum trajectory strength is identified and the estimation that this trajectory provides for the k^{th} frame is chosen as the period estimate. The HR estimate is then obtained from the period estimate using the equation 2.1

CHAPTER 3
Present Work : Real Time Implementations

3.1 Previous Work

Currently, there exists an Android Application - PULSE developed by Pedro Boloto Chambino at Farunhofer Portugal[20]. The PULSE application estimates the Heart Rate of a person using the EVM Framework. This application performs a real-time EVM and estimates the heart rate using the device's camera. In this application, first the image is captured from the device's camera. After capturing the image, a person's face is detected using the OpenCV object detect module which has been previously trained to detect human faces. This Region of Interest (ROI) i.e. the person's face is then fed into the implemented EVM to amplify colour variations. The average of the green channel is computed and stored. It is proposed that the stored values represent a PPG signal of the underlying blood flow variations. It is then validated as cardiac pulse signal by detecting its peak in order to verify its shape and timing (Pulse onset detection). The Heart rate estimate is then computed by identifying the frequency with the highest magnitude in the power spectrum of the signal. Power Spectrum is calculated as the squared magnitude of the Fourier transform of a signal. Since the values are captured from a video which is a discrete sequence of images, the signal under consideration is discrete, with a sampling frequency equal to frames per second (fps). The index, i, corresponding to the maximum of the power spectrum can then be converted into a frequency value F, using the equation

$$F = (i \times FPS)/2N \tag{3.1}$$

where N is the size of the signal extracted. The EVM implementation carried out in this application is not exact i.e. it is a light-weight implementation with a reduced computational cost. Due to these reasons this application is highly erroneous. In the next section we shall elaborate on the technologies utilized and the implementation details of our Android Application, which overcomes the above mentioned drawbacks and provides more accurate heart rate estimates.

3.2 Technologies

3.2.1 Android Software Development Kit (SDK)

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android smartphones are the most used smartphones in the world. Native Android apps are developed using the Java programming language, and can easily be ported to other mobile operating systems like Blackberry, Symbian and Ubuntu. In addition, Android apps can also be ported easily to Chrome OS. Android SDK is the development kit for the Android platform. Because of its open source code and permissive licensing, it allows the software to be freely modified and distributed. This has allowed Android to be the software of choice for technology companies that require a low-cost, customizable, and lightweight operating system for mobile devices and others. Android has also become the largest installed base of all operating systems of any kind [16]. Android consists of a kernel based on Linux kernel with middleware, libraries and APIs written in C. Applications, usually run on an application framework which includes Java-compatible libraries. Java bytecode is then translated to run on the Dalvik virtual machine. Support for simple C application is possible, by the usage of Java Native Interface (JNI), a programming framework that allows Java code to call and be called by libraries written in C/C++. JNI has been employed in our Android platform implementation.

3.2.2 OpenCV - Computer Vision Library

OpenCV stands for Open Source Computer Vision. It is a library of programming functions mainly aimed at real time computer vision and image processing applications. This library is cross-platform and free for use and modification under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though the older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. All of the new developments and algorithms in OpenCV are now developed in the C++ interface [17]. Our present implementation is also developed in C++ with extensive usage of OpenCV libraries in Android platform.

3.3 Implementation Details

3.3.1 Overview

Mode based period estimation method was implemented in Android platform. The application workflow is as follows

1. *Frame Acquisition* : Read the original frame from device camera.
2. *Video Processing* : Captured frame is then processed as explained below.
   (a) *Face Detection* : Detect face in each frame.
   (b) *Eulerian Video Magnification* : Colour magnify detected region of interest.
   (c) *Points Extraction* : Select 20 points on the magnified face.
   (d) *Pixel Tracking* : Track the pixel intensity values over time from the selected points.
   (e) *Mean Removal* : Tracked values are subjected to mean removal to remove DC value.
   (f) *Autocorrelation* : Period is estimated in each of the tracked 20 points using autocorrelation.
   (g) *Median Filter* : The autocorrelation time-series is smoothed using a median filter.
   (h) *Mode* : Mode of the estimated periods is determined.
3. *Validation and Heart Rate Estimation* : HR estimate in Beats Per Minute (BPM) is obtained from calculated mode if the obtained mode is in the valid range.

The above procedure is repeated for every frame captured from the device camera.

3.4 Face Detection

Face detection step uses OpenCV implementation of cascaded object detector, which has been previously trained to detect faces in images [18] and [19]. Because object detectors are computationally expensive and for performance improvement, a minimum size for the face detector was set to 40% of the frame width and height. Face detection was done once in every 0.5 seconds as the subject under consideration is assumed to be in rest. Eulerian Video Magnification was carried out only on the detected face rectangle instead of the entire frame to reduce computation as well as extraneous artefacts arising from the parts of the frame which does not contain face.

3.5 Eulerian Video Magnification Implementation

EVM implementation was done in C++ as it reduces the number of JNI calls from the Android JVM and increases the application performance. EVM mainly consists of four operations and their implementations is discussed in this section.

1. **Resize Down** : This step applies a spatial filter by calculating the level of the Gaussian pyramid. This is achieved by looping to the desired level where the input to the next loop is the result from the previous loop, starting with the original frame. A Gaussian pyramid level is calculated by, first, convolving the input frame with the kernel and then, downsampling the frame by rejecting even rows and columns. This step was implemented using the OpenCV interpolation method named area, which produces a similar result to the one provided by using a Gaussian filter and downsampling the image. However, instead of, iteratively downsampling the frame multiple times, it is now a single resize to a predefined size.

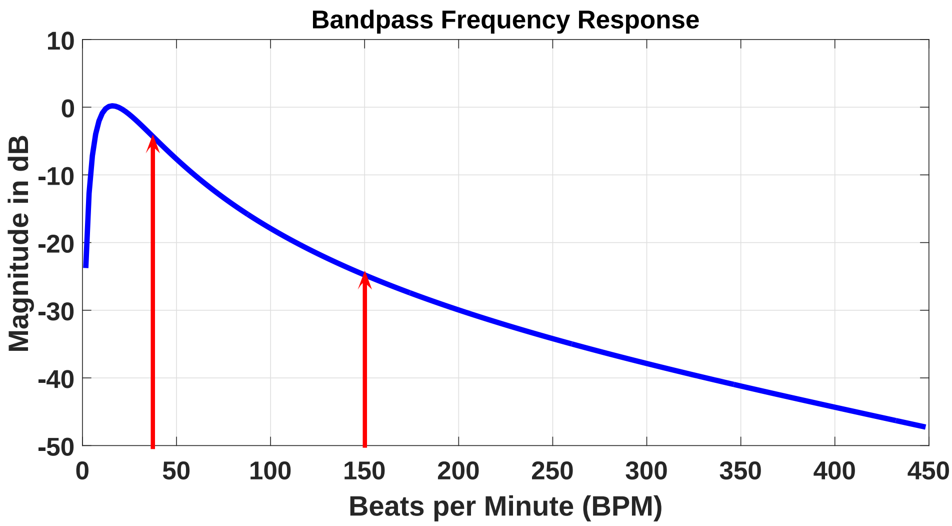


Figure 3.1: Frequency Response - Temporal Bandpass Filter

2. **Temporal Filter** : In the MATLAB implementation of EVM an ideal temporal IIR bandpass filter was employed. However in real time implementation an ideal filter is computationally expensive, therefore a lightweight bandpass filter was implemented.

The filter was constructed from the subtraction of two first-order lowpass IIR filters. Each lowpass filter is computed as follows:

$$F_n = F_{n-1} * (1 - \omega) + \omega * M \tag{3.2}$$

where M is the current frame, F is the lowpass filter accumulator for each frame, and ω is the cutoff frequency percentage($\omega = F_c/(F_s/2)$). Here F_c is the cutoff frequency and F_s is the sampling frequency and $F_s/2$ represents maximum frequency component.

Figure 3.2 shows the frequency response of implemented temporal bandpass filter with cutoff frequencies 40 BPM and 120 BPM. It can be inferred from the figure 3.2 that the lightweight implementation is an approximate bandpass filter.

3. **Amplification** : This step multiplies an amplification factor α, which results in the magnification of the colour variation selected by the temporal filter.

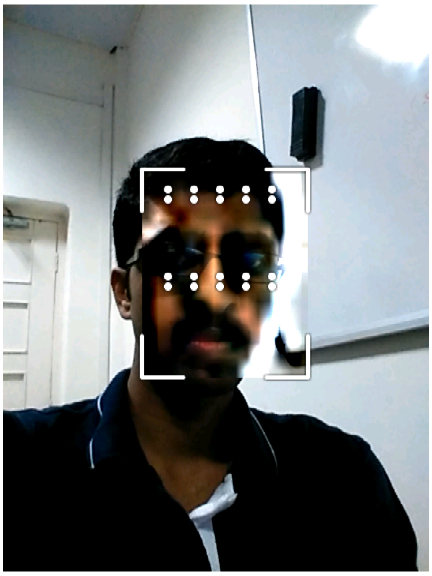


Figure 3.2: Screen shot of the amplified video

4. **Resize Up** : This step performs the inverse operation of the resize down step, where it upsamples the frame by inserting even rows and columns with zeros, and then, convolves the input frame with the same kernel as in the resize down step. This step was also modified to a single resize operation using the linear interpolation method, which produces a similar result to the one used previously where the image was upsampled iteratively using a Gaussian filter.

3.6 Validation and Heart Rate Estimation

Autocorrelation is performed on the the signals obtained by tracking intensity values of the selected 20 points on the face. The most repeated period value (mode) estimated from autocorrelation is used for heart rate calculation. If the estimated heart rate does not lie in 40 BPM to 150 BPM range, this value is ignored as the heart rate of the subject under rest cannot exceed this range. In such cases, the second most repeated value is used for estimating heart rate. In situations where even the second most repeated value also does not fall in that range, the estimate obtained for previous frame is used as the estimate for this frame as well.

The tracked intensity values are stored in a matrix of fixed size. The matrix functions as a stack, when the matrix is filled, the first data captured is moved out of the matrix to accommodate for the new incoming value. In our present implementation the size of the matrix was fixed to 100 samples.

3.7 Application Features

- *Magnification Factor* α : Users can vary the magnification factor of the EVM framework.
- *Average Heart Rate* : Users will be shown their average Beats per Minute.
- *Frames per second display* : Users can also view the frame rate at which their face video is being captured.
- *Heart Rate from hand* : Users can get to know their heart rate from their hand video which can be captured using back camera.

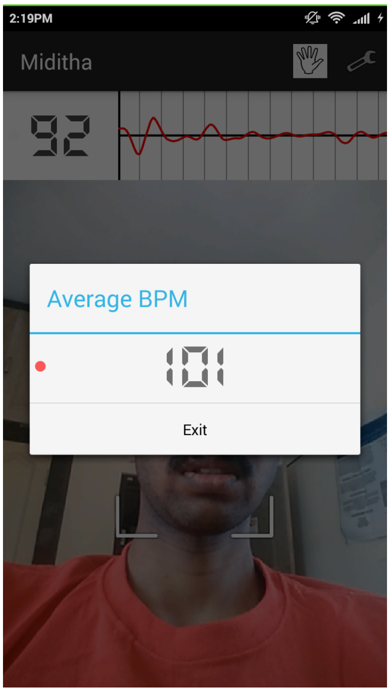


Figure 3.3: Screenshot showing average Heart rate in Beats per minute

CHAPTER 4
Results and Conclusions

4.1 Results : Matlab Simulations

- Absolute average error (AAE) - metric to evaluate performance

$$AAE = \frac{1}{F}\sum_{f=1}^{F}|BPM_{est}(f) - BPM_{true}(f)| \tag{4.1}$$

F - number of frames, BPM_{true} - Ground truth HR from ECG, BPM_{est} - Estimated HR

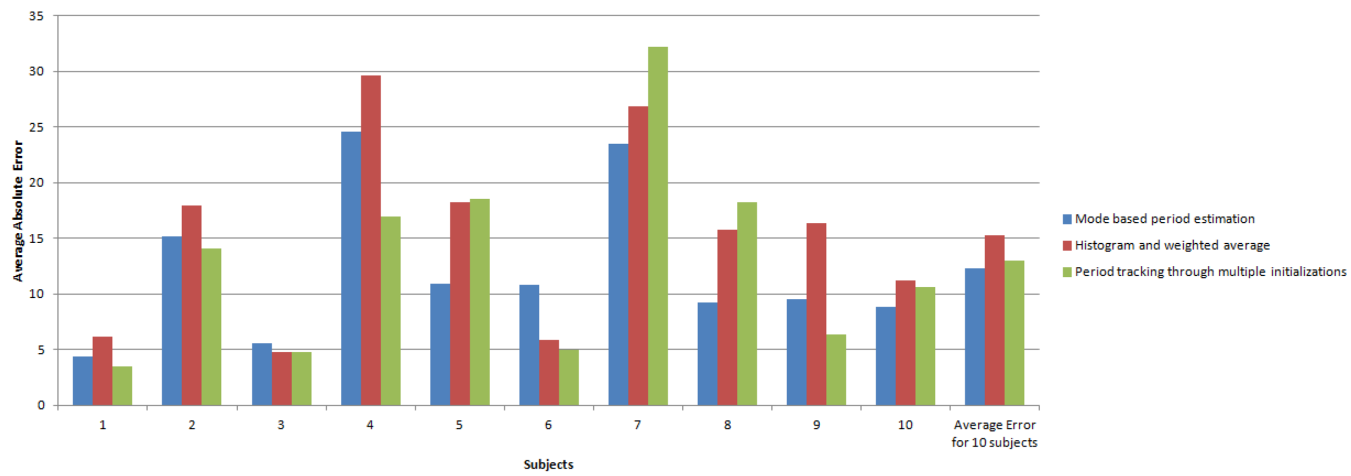


Figure 4.1: Performance of various estimation methodologies

Performance analysis was done by calculating AAE between Estimated HR and the Ground truth obtained using Finger tip Pulse-Oximeter, for all the 10 subjects. Below, we have shown the estimated and Ground truth for subject 1.

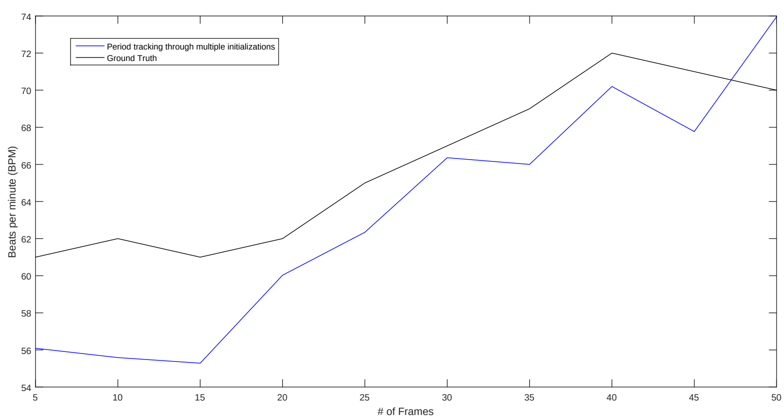


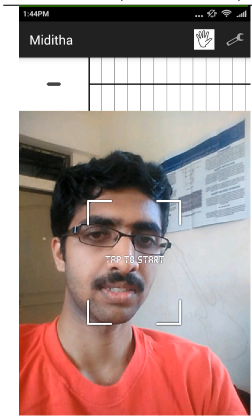
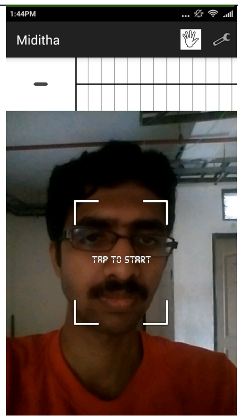
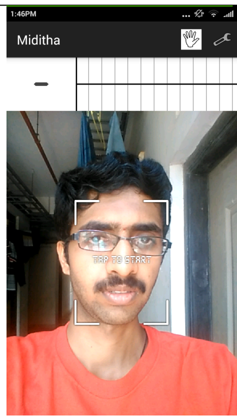
Figure 4.2: Comparison of estimated HR and Ground Truth for subject 1

4.2 Results : Real Time Implementation

4.2.1 Baseline Analysis

Subjects	Ground Truth	Baseline-1	Baseline-2	Miditha
1	**75**	62	60	80
2	**83**	55	60	78
3	**83**	67	67	76
4	**73**	55	72	70
5	**67**	57	73	63

4.2.2 Lighting Analysis : Ground Truth = 94 BPM

Room (Medium)	Dark	Bright
		
99 BPM	77 BPM	89 BPM

We observe that brighter lighting conditions provide more accurate estimates of heart rate.

4.2.3 Heart Rate from Hand

Heart Rate estimate from hand video can also be obtained using the hand icon provided in the toolbar of the application.

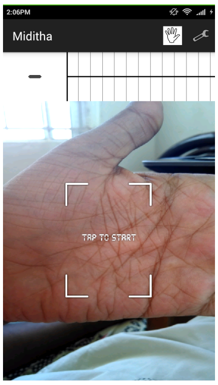


Figure 4.3: For a Ground Truth of 94 BPM, the application gave an estimate of 89 BPM

4.3 Conclusions

Our application performs better than existing heart rate estimator from face video. It also provides heart rate estimate from hand video. We observe that the HR estimates are sensitive to frame-rate of the camera. Developed application has been tested under different lighting conditions and analysed. Also, Pulse transit time which is required for Blood pressure calculation was estimated.

Verifying the estimated pulse transit time obtained from face and wrist videos is part of future work. We also intend to upload the developed android application to Google Playstore.

REFERENCES

[1] S. Cook, M. Togni, M. C. Schaub, P. Wenaweser, and O. M. Hess, “High heart rate: a cardiovascular risk factor?” Eur. Heart J. 27(20), 23872393 (2006).

[2] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, “Remote plethysmographic imaging using ambient light” Opt. Expr., vol. 16, pp. 2143421445, Dec. 2008.

[3] M. Z. Poh, D. J. McDuff, and R. W. Picard, “Non-contact, automated cardiac pulse measurements using video imaging and blind source separation” Opt. Expr., vol. 18, pp. 1076210774,May 2010.

[4] Liu, C., Torralba, A., Freeman, W. T., Durand, F., and Adelson, E. H. 2005. “Motion magnification” ACM Trans. Graph. 24, 519526.

[5] S. Ulyanov and V. Tuchin, “Pulse-wave monitoring by means of focused laser beams scattered by skin surface and membranes,” in Proc. SPIE, Los Angeles, CA, 1884, pp. 160167.

[6] E. Greneker, “Radar sensing of heartbeat and respiration at a distance with applications of the technology” in Proc. Conf. RADAR, Edinburgh, U.K., 1997, pp. 150154.

[7] M. Garbey, N. Sun, A. Merla, and I. Pavlidis, “Contact-free measurement of cardiac pulse based on the analysis of thermal imagery” IEEE Trans. Biomed. Eng., vol. 54, no. 8, pp. 14181426, Aug. 2007.

[8] J. Fei and I. Pavlidis, “Thermistor at a distance: Unobtrusive measurement of breathing” IEEE Trans. Biomed. Eng., vol. 57, no. 4, pp. 988998, Apr. 2009.

[9] J. Allen, “Photoplethysmography and its application in clinical physiological measurement” Physiol. Meas., vol. 28, pp. R1R39,Mar. 2007.

[10] Poh, Ming-Zher, Daniel J. McDuff, and Rosalind W. Picard. “Noncontact, automated cardiac pulse measurements using video imaging and blind source separation” Optics Express 18 (2010): 10762. 2011 Optical Society of America.

[11] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frdo Durand, William T. Freeman “Eulerian Video Magnification for Revealing Subtle Changes in the World” ACM Transactions on Graphics, Volume 31, Number 4 (Proc. SIGGRAPH), 2012

[12] Wang R, Jia W, Mao Z-H, Sclabassi RJ, Sun M. “Cuff-Free Blood Pressure Estimation Using Pulse Transit Time and Heart Rate. International conference on signal processing proceedings International Conference on Signal Processing” 2014;2014:115-118. doi:10.1109/ICOSP.2014.7014980.

[13] http://people.csail.mit.edu/mrub

[14] Lakshminarasimha Murthy, N.K.; Madhusudana, P.C.; Suresha, P.; Periyasamy, V.; Ghosh, P.K., "Multiple Spectral Peak Tracking for Heart Rate Monitoring from Photoplethysmography Signal During Intensive Physical Exercise," in Signal Processing Letters, IEEE , vol.22, no.12, pp.2391-2395, Dec. 2015

[15] Zhilin Zhang; Zhouyue Pi; Benyuan Liu, "TROIKA: A General Framework for Heart Rate Monitoring Using Wrist-Type Photoplethysmographic Signals During Intensive Physical Exercise," in Biomedical Engineering, IEEE Transactions on , vol.62, no.2, pp.522-531, Feb. 2015

[16] https://en.wikipedia.org/wiki/Android_(operating_system)

[17] https://en.wikipedia.org/wiki/OpenCV

[18] Paul Viola and Michael 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 on, volume 1, pages I511. IEEE, 2001.

[19] Rainer Lienhart and Jochen Maydt. "An extended set of haar-like features for rapid object detection", In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I900. IEEE, 2002.

[20] Pedro Chambino. "Android-based implementation of Eulerian Video Magnification for vital signs monitoring", PhD Dissertation