Mobile Pulsecam: Blood Perfusion Imaging Using Dual-Camera Smartphones

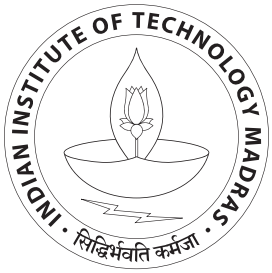
A Project Report

submitted by

ANIRUDTH N

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.
May 2019

THESIS CERTIFICATE

This is to certify that the thesis entitled **Mobile Pulsecam: Blood Perfusion Imaging Using Dual-Camera Smartphones**, submitted by **Anirudth N** (**EE15B007**), to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelors of Technology** is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Kaushik Mitra
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Prof. Ashok Veeraraghavan
Research Co-Guide
Associate Professor
Dept. of Electrical & Computer Engineering
Rice University, TX 77005, USA

Place: Chennai

Date:

ACKNOWLEDGEMENTS

This work would not have been possible without the guidance and the help of several people. I take this opportunity to extend my sincere gratitude to all those who made this thesis possible.

First, I would like to thank all my teachers who bestowed me with good academic knowledge. I am very indebted to my guide, Prof. Kaushik Mitra, for his generous guidance and support. Despite the project sounding very ambitious in the start, it was his belief in the idea and investment, both in terms of time and money, which turned it successful. I am also very indebted to my co-guide, Prof. Ashok Veeraraghavan (Rice), for creating time in his busy schedule to clear my doubts and help me overcome all the hurdles. I would also like to thank Akash Kumar Maity (Rice) and Amruta Pai (Rice) for sharing their valuable insights and code bases.

Lastly, I would like to thank my family for giving support and guidance all through my life.

ABSTRACT

KEYWORDS: Blood Perfusion, Smartphone-Based Diagnostics, Biomedical Imaging, Computer VIsion

Blood perfusion is the rate of flow of blood to a particular tissue in the body. Measuring blood perfusion is vital in ensuring adequate oxygen delivery for important functioning of various organs. Measuring superficial blood perfusion of the skin acts as an indicator in understanding of general blood flow and also aids in the understanding of the beating of the heart, diabetic levels, fat levels, obstacles to blood flow etc. Classically, blood perfusion is measured using costly, bulky Laser Doppler or Speckle Imaging machines. Rice University had come up with a cheap, portable alternative called Pulsecam, which would produce blood perfusion maps using just a camera and a pulse oximeter. We present a dual-camera based blood perfusion approach born out of Pulsecam, called Mobile Pulsecam. Our major contributions include the world's first smartphone-based approach for blood perfusion estimation, a real-time skin blood perfusion estimator and an Android app to achieve the same. We strongly believe that this would allow remote healthcare access to all.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Social Perspective

Medical innovation in India has been making giant leaps over the past few years Gangolli *et al.* [2005]. There have been new inventions in drug development and vaccinations along with marked improvements in diagnostics. We are now able to diagnose diseases in their infant stages and eliminate the cells/microbes even before they multiply. However, a lot of these improved inventions are not accessible to the entire population Rao and Mant [2012].

Improving medical accessibility has been worked out in our country more in the form of logistics. There have been apps to reach out to doctors in emergency situations and get medicines delivered at home. There hasn't been a technological healthcare breakthrough for medical accessibility. Granted, we have been inventing new medical tests. However, the reason why these have not lead to a major disruption is because many of the machines needed to do such testing are bulky and costly. That means only hospitals with good financial backing and placed at a suitable location have access to such expensive machines. In a country with a sizable portion of the population living under the poverty line, that translates to people having to shell out a good amount of money to get these tests done. Hospitals Binnendijk *et al.* [2012] situated in remote locations, like the north-east, often have difficulties even in procuring such machines.

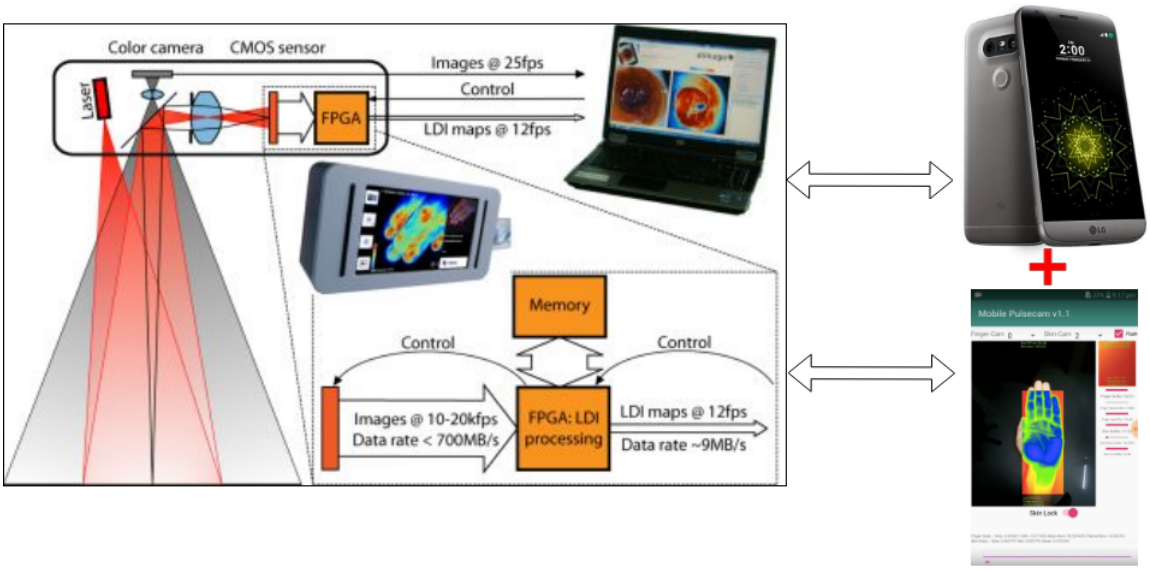


Figure 1.1: In a Nutshell: Aiming for Laser Doppler Imaging through a Smartphone App [Credits: Google Images]

On the other hand, smartphone penetration has been increasing at an exponential rate. By the end of 2019, 1.2 billion Indians are poised to be using 5 billion smartphones. Smartphones nowadays have multiple cameras and sensors including the capability to generate ECGs! Clearly, smartphones are the best way to reach the public masses, indeed if we're able to generate a medical testing application with it.

1.2 The Technical Perspective

Building low-cost alternatives to bulky medical machines has been an active area of research in the past decade. There has been a recent work in Rice University Kumar *et al.* [2016] where they replicate the blood perfusion tests of Laser Doppler and Speckle imaging techniques using a camera and a pulse oximeter. Blood perfusion is the rate of flow of blood to a particular tissue through blood vessels in the body. They focus mainly on estimating surfacial blood perfusion on the skin.

Though the camera and pulse oximeter setup tries to bring down the costs, the setup has to be administered by a trained personnel. In our work, we aim to replace the pulse oximeter in this algorithm with a secondary camera and make the entire algorithm function within a compact smartphone application.

The major contributions of this project include

1. The world's first smartphone-based approach for blood perfusion estimation
2. A dual-camera technique to extract blood perfusion
3. A real-time algorithm for blood perfusion estimation
4. An easy-to-use smartphone application which performs the entire testing

This thesis first starts with a Literature Review. This is followed by a chapter devoted to explaining the original Pulsecam algorithm followed by a chapter on the modifications made to get it to work on a smartphone. These are trailed by a Results and a Conclusions chapter.

CHAPTER 2

Literature Review

Blood flow (or perfusion) is vital in ensuring oxygen delivery to the cells and in maintaining metabolic homeostasis. Measuring peripheral perfusion, i.e. perfusion of the blood just underneath the skin surface is important in both medical and surgical fields Allen and Howell [2014] including assessment of peripheral perfusion in critical care, tissue viability in plastic, reconstructive, and burn surgery Corstian *et al.* [2008] as well as for wound assessment. Blood perfusion generally varies from one tissue site to another, and can also change over time due to varying metabolic demands, and so a spatial map of blood perfusion over time, i.e. a three-dimensional quantity, is usually measured. By measuring blood perfusion, one could estimate heart rate variability, blood pressure etc. Infact, using machine learning on spatial blood perfusion maps, one could non-invasively even estimate the blood sugar levels without actually pricking for a sample of blood!

Laser speckle contrast imaging Forrester *et al.* [2004] and laser Doppler imaging Wardell *et al.* [1993] Serov *et al.* [2005] devices are the de-facto methods of measuring blood perfusion. However, these devices are (i) costly and bulky, (ii) can cause interference with other instruments, (iii) lead to discomfort for patients over long exposure, and (iv) are too expensive to be used for daily care. Some single camera blood perfusion imaging techniques Kamshilin *et al.* [2013] Kamshilin *et al.* [2011] were proposed. The camera is focused on our region of interest on the skin where we want to estimate the blood perfusion. However, the main drawback with these are noisy and low-resolution maps because small motion or illuminance difference

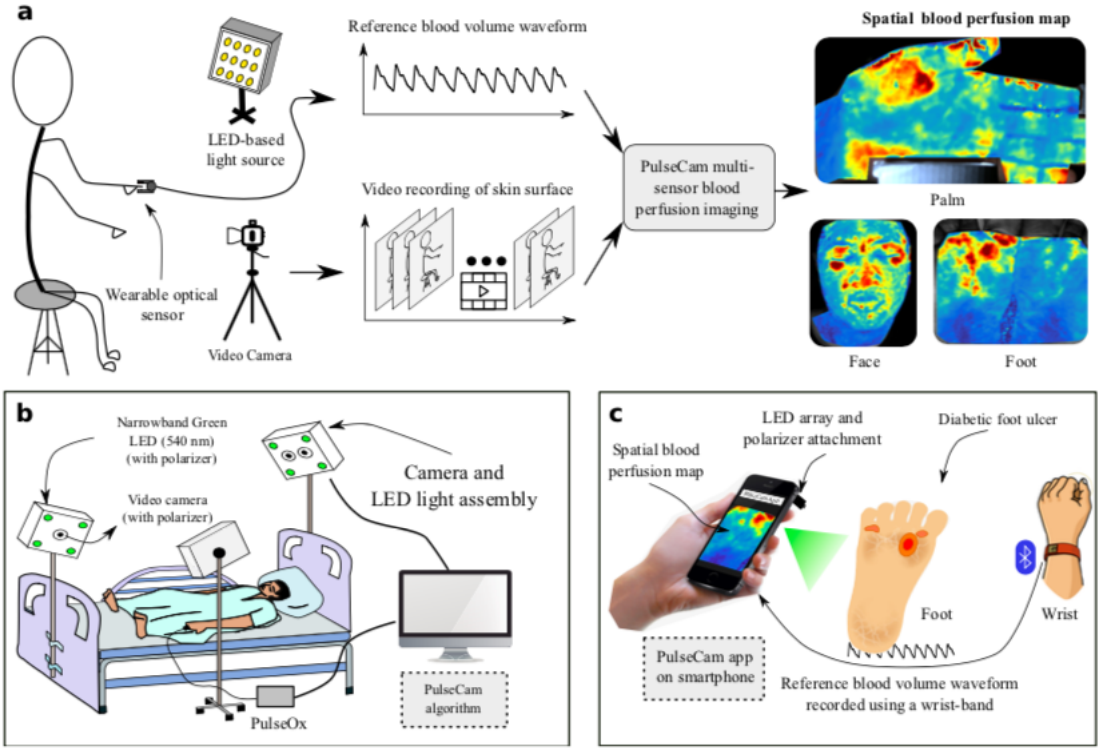


Figure 2.1: A schematic overview of Pulsecam as shown in Kumar *et al.* [2019]

would throw up as artifacts in the perfusion estimation. Some of these are real-time like Rubins *et al.* [2011].

Pulsecam Kumar *et al.* [2016] overcomes these difficulties by performing sensor fusion between a camera and a pulse oximeter. A pulse oximeter is a simple spot measurement device which can measure blood volume waveform reliably from one body location, for example a finger or toe. This would give a high reliable, less noisy reference signal for the blood volume waveform at that point. Since blood flows serially through out the body, the pattern of the blood volume waveform captured by the pulse oximeter would be the same as that captured by the camera. Although, there might be a slight delay (10ms) in the propagation of the waveform from the pulse oximeter region to the camera-captured region. With a proper physiological model, one could use the blood volume waveform from the pulse oximeter as a good reference and estimate high-accurate high-resolution

spatial perfusion maps over the skin region of interest. This is basically what Pulsecam does. A schematic overview of the Pulsecam paper is shown in Figure 2.1. The physiological model developed in Pulsecam is similar to their other novel PPG estimation algorithm, DistancePPG Kumar *et al.* [2015].

It is also shown in literature Gregoski *et al.* [2011] about camera-based pulse oximeters in smartphones. Here, the blood volume waveform is measured placing a finger illuminated by the phone's flash on the camera. From this, it is shown that one could accurately estimate the blood volume waveform.

In our work, we replace the pulse oximeter in Pulsecam Kumar *et al.* [2016] with a secondary camera where the user would place her finger. Through this, we create a virtual pulse oximeter. We aim for our algorithm to run on a generic dual camera smartphone: one camera would focus on the skin region of interest and another will capture PPG from a finger. Through this, we perform sensor fusion similar to Pulsecam to get high-resolution spatial blood perfusion maps. We subsequently modify the Pulsecam algorithm in order to run it in real-time within the computational capabilities of a smartphone.

Hence, our project is aptly named **Mobile Pulsecam**.

CHAPTER 3

Pulsecam Formulation

This chapter will introduce the mathematical modeling behind Pulsecam. Furthermore the estimation of blood perfusion from only image data is delineated, inspired from Kumar *et al.* [2016]. A high-level picture is shown in Figure 3.1.

3.1 Blood Perfusion Signal And Noise Model

Blood perfusion (or flow) is the rate of change of blood volume at a particular location over time. The volume of blood in the vessels is in sync with heart beat. As the heart rate remains more or less constant towards equilibrium, the blood perfusion remains proportional to the amplitude of the blood volume waveform. Hence, in order to measure the blood perfusion at a particular region of spatial interest, one has to estimate the amplitude of the blood volume waveform at the particular tissue location over time.

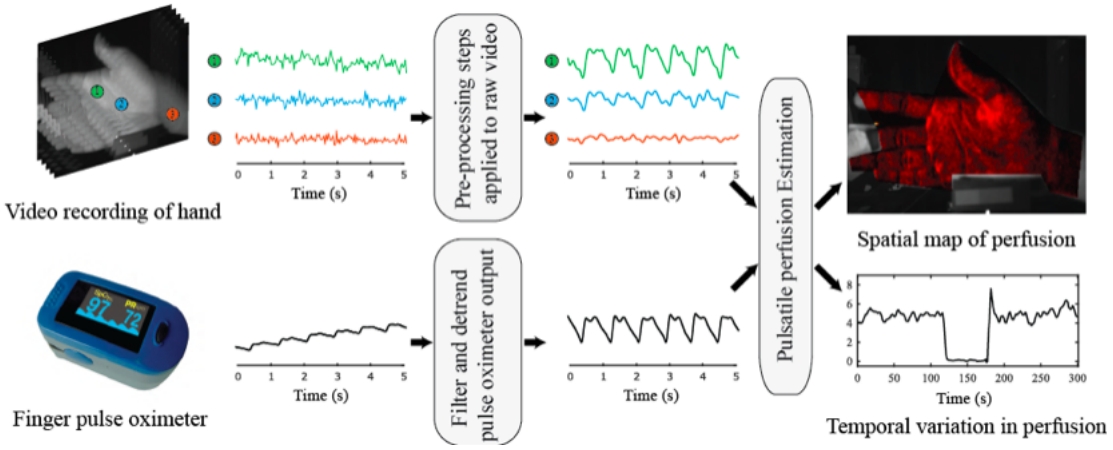


Figure 3.1: An overview of Pulsecam working Kumar *et al.* [2016]

Let the skin region of interest, over which we would like to estimate blood perfusion, be illuminated by a uniform light source. When light falls on the skin surface, part of it is partially reflected and part of it is absorbed by the skin and the underlying tissue. The part of light which is reflected back is recorded by the image sensor.

Let $\vec{x}$ denote a specific coordinate of interest on the image plane. Let t denote a particular time instant. Let λ denote the specific wavelength of interest. We assume that the incident background lighting of the skin surface remains constant over time. Let this be denoted by $L_\lambda(\vec{x})$. This can be thought of coming from a constant light source. The camera-captured intensity includes both a surface and sub-surface reflectance component. Let the surface component be denoted as $b_\lambda(\vec{x}, t)$ and the sub-surface component be denoted as $c_\lambda(\vec{x}, t)$. Let the image intensity captured by the camera be represented as $I_\lambda(\vec{x}, t)$. Then the blood perfusion signal model can be expressed as

$$I_\lambda(\vec{x}, t) = L_\lambda(\vec{x})(b_\lambda(\vec{x}, t) + c_\lambda(\vec{x}, t)) + w(\vec{x}, t) \tag{3.1}$$

where $w(\vec{x}, t)$ represents the camera's measurement noise.

A further approximation can be made here. The surface reflection component behaves like a mirror-like reflection of incident light. It can also be expressed as a function of the amount of incident light absorbed by skin tissue and cells underneath. If the skin region is assumed to be stationary throughout the capture time, then

$$b_\lambda(\vec{x}, t) \approx b_\lambda(\vec{x}). \tag{3.2}$$

i.e the surface reflectance component behaves time-invariantly.

The sub-surface reflectance component $c_\lambda(\vec{x}, t)$ is due to light absorption by blood chromophores and is time varying due to pulsatile changes in the blood volume in the microvasculature underneath the skin surface. It can show spatial, temporal and spectral variations. The spectral dependence is due to wavelength dependent variations in light absorption by hemoglobin and oxy-hemoglobin in the blood as well as variations in the penetration depth of light of different wavelengths. The subsurface reflectance component can be decoupled as

$$c_\lambda(\vec{x}, t) = a_\lambda(\vec{x}, t)p(t - \tau(\vec{x})) \tag{3.3}$$

where $a_\lambda(\vec{x}, t)$ is the amplitude of the blood volume waveform $p(t)$ and is different at different location. The amplitude $a_\lambda(\vec{x}, t)$ can also change over time due to temporal variations in blood perfusion, but at a rate that will be much slower in comparison to the instantaneous variations in $p(t)$ (which is in sync with the beating of the heart). Here, $\tau(\vec{x})$ represents the time-delay of the waveform as it propagates through the arterial network. However, for small regions of interest, such as the palm or foot, it can be approximated as a constant i.e $\tau(\vec{x}) \approx \bar{\tau}$. If we also assume the illumination to be constant across the region of interest, the overall simplified model of blood perfusion in our case becomes

$$I_\lambda(\vec{x}, t) = L_o(b_\lambda(\vec{x}) + a_\lambda(\vec{x}, t)p(t - \tau(\vec{x}))) + w(\vec{x}, t) \tag{3.4}$$

3.2 Pulsecam Blood Perfusion Estimation

Blood perfusion estimation using Pulsecam can be decomposed into a series of three preprocessing steps followed by a fusion stage of the camera and pulse oximeter readings to obtain the blood perfusion map.

3.2.1 Preprocessing

1. As a first step, apply a $M \times M$ mean filter on the video image of the region of interest. This helps in reducing the camera noise contribution. However, there is a trade-off between output resolution and SNR when it comes to choosing M. It is shown in Kumar *et al.* [2016] that $M = 4$ or $M = 5$ performs well.
2. In the second step, the spatially averaged video is filtered, temporally at each pixel, using a bandpass filter having passband between 0.5Hz to 5Hz. This is the frequency range over which the heart waveforms lie. Simultaneously, the pulse oximeter waveforms are also filtered temporally using the same filter. The output at this stage is $I_{AC}(\vec{x}, t)$ and $p_{AC}(t)$.
3. In order to remove scaling effects, because of the background factor in incident light intensity, one method proposed is to use a grayscale chart, calibrate for L_o and divide from $I_{AC}(\vec{x}, t)$. Another method is to estimate the envelope of $I_{AC}(\vec{x}, t)$ and divide by the corresponding amplitudes at each t, to restrict the magnitude of the envelope to 1. A third method is to low pass filter $I_{AC}(\vec{x}, t)$ across time, and divide the high-pass outputs by the low-pass outputs accordingly. Let the normalized $V_{AC}(\vec{x}, t)$ be represented as $V_N(\vec{x}, t)$.

3.2.2 Fusion

If we assume that the noise in the camera measurement is white and Gaussian, then the maximum likelihood estimator for the perfusion $a_\lambda(\vec{x}, t)$ is

$$\hat{a}_{ML}(\vec{x}, t) = \max_{\tau(\vec{x})} \langle V_N(\vec{x}, t - \tau(\vec{x})), p(t) \rangle_T \tag{3.5}$$

where $\langle ., . \rangle$ is the inner product between the two vectors over a time interval

T. $\hat{a}_{ML}$ is the normalized blood perfusion estimate. The time window T is chosen in such a way, such that the perfusion estimate remains constant over this time (typically 10 ms). Referring to Kumar *et al.* [2016], readout or thermal noise generally follows a Gaussian distributed, and under sufficient illumination, cameras shot noise also follows a Gaussian distribution. Quantization noise is uniformly distributed, but if we assume a minimum of 10 - 20 pixel in a pixel block over which spatial averaging is done (M = 4), then due to central limit theorem, quantization noise can also be modeled as Gaussian. Noise due to motion artifact is generally spiky and difficult to model, and is not considered here.

The signal-to-noise ratio of the above ML estimate is same as the SNR of the signal of interest and can be estimated as

$$SNR = \frac{\hat{a}^2_{ML}(\vec{x}, t)}{\mathbf{Var}(V_N(\vec{x}, t) - \hat{a}_{ML}(\vec{x}, t)p(t - \tau(\vec{x}))} \tag{3.6}$$

CHAPTER 4

Mobile Pulsecam Implementation

The algorithm of Pulsecam has been described in the previous chapter in the classical sense. However, in order to arrive at an implementation of this in a smartphone is very non-trivial. In this section, the proposed modifications to the algorithm are detailed. This chapter will start with a high-level vision, followed by modifications and implementation specifics.

4.1 The Big Picture

A high-level vision for this project is to enable maximal remote disease diagnosis. That means this application must be easily usable and inferable by an average human being. In order to fine tune the implementation of the algorithm, we first create the big-picture overview of the way we would like it to be used. A sketch of the user workflow can be seen in Figure 4.1. A more detailed description is given below.

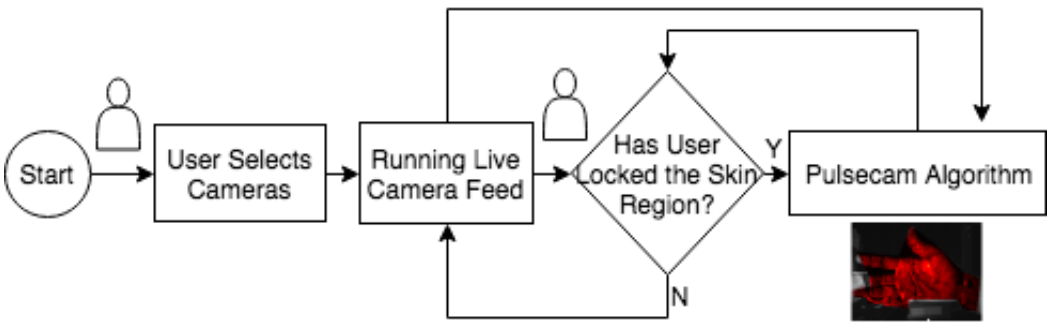


Figure 4.1: User Interactive Architecture of Mobile Pulsecam

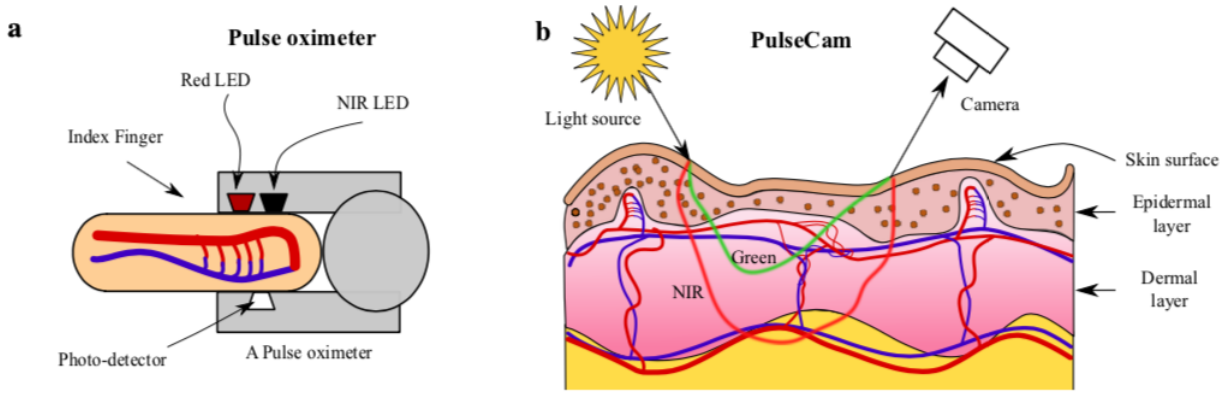


Figure 4.2: Pulse Oximeter versus Virtual Pulse Oximeter working Kumar *et al.* [2019]

4.1.1 App Workflow

- The user opens the app and chooses the two cameras, one for acting as a pulse oximeter and another acting as a camera. The unsynchronized live camera feed then starts.
- The user shows the interested skin region in the camera. A skin detector would draw a bounding box around this region of interest. If the region of interest is correctly bound-boxed, user then locks the skin region.
- A tracker will then shadow and track the region of interest throughout the video, and pass this along with the pulse oximeter camera's reading to the Mobile Pulsecam algorithm
- The output of the Mobile Pulsecam algorithm is overlayed over the region of interest in the live camera feed.
- When the user switches off skin lock, the blood perfusion estimating loop comes to a halt.

4.2 Modifications

Given the vision for Mobile Pulsecam in the previous subsection, the following are some modifications which were made in order to accommodate a real-time blood perfusion stream.

4.2.1 Virtual Pulse Oximeter

In the original Pulsecam formulation, a pulse oximeter was used to generate the reference waveform. However, many apps in the recent smartphone era like have shown to estimate the blood volume waveform from placing a finger over a camera. This technique is even competitive enough to an original pulse oximeter. There is a fundamental difference here, in the way the two devices operate.

In case of a pulse oximeter, light projected from under the finger is captured by a sensor imaging the top of the finger as shown in Figure 4.2. Whereas in case of our virtual pulse oximeter, the rear flash of the smartphone is switched on. The light reflected from the finger is captured by our camera sensor. As a result, our setup captures the surface variation of the blood volume waveform which is very important for us. Infact, in a recent Nature preprint Kumar *et al.* [2019], it is even being shown that our virtual pulse oximeter formulation is more robust to blood volume artifacts from deep arteries.

Once we have frames coming from the camera feed of the finger, we just perform an averaging over the entire frame matrix. Further processing based on this metric seems to give a good reference estimator for the blood volume waveform in our experiments.

4.2.2 FIR Filters

The original Pulsecam algorithm, detailed in Chapter 3 uses classical butterworth filters for performing filtering operations. These IIR filters require an infinite buffer collection of data points for filtering. However, this is not feasible in our case. Hence, we shift to Finite Impulse Response filters designed using the Remez

Exchange Algorithm McClellan and Parks [2005].

4.2.3 Run Time

The original Pulsecam algorithm is not designed to run in real-time. If we were needed to run the original algorithm offline in Android after capturing inputs, it was estimated to take about 5-10 minutes.Our app doesn't have permissions to do heavy lifting in the background. If the user changes to another app, Mobile Pulsecam will be killed from the memory. It wouldn't also be practical for the user to stay idle for about 5-10 minutes to see the outputs. We also wanted the user to be able to appreciate the perfusion waveform variation over their skin region of interest in real-time. Hence, we prioritized for a real-time implementation and the algorithm was suitably modified to achieve this.

4.3 Concise Mathematical Model

Refer to Section 3.1 for a rigorous mathematical treatment of Pulsecam. In this subsection, we focus more closely on the implementation aspect of math. A feel about various hyperparameters is also conveyed. Also, some of the math here might be familiar to Kumar *et al.* [2019], although this work has been performed independently.

Note that absorption characteristics of Haemoglobin, present in the blood, are seen to be maximum in the green wavelength region of the light spectrum. Hence, for pulsecam computations, we only use the green channel images as inputs to the algorithm.

Let the frame rate at which both the cameras run be f Hz. In almost all of our experiments, $f = 20Hz$ gave best results.

Let $P_F[n; i, j]$ be the video stream coming from the finger camera and let $P_S[n; i, j]$ be the video stream coming from the skin region-of-interest camera. Here, n is an index of sample time and i, j represent pixel positions within a frame. The pixel values are normalized between 0.0 to 1.0 from 0 to 255. Furthermore, lets assume that both of these video streams are synchronized and exactly sampled at f Hz. That means the parameter $\tau(\vec{x})$ mentioned in Section 3 in Pulsecam model is negligibly small in this case. Hence, we ignore this parameter. The SpO2 blood waveform estimator from finger is given as $\hat{p}_{ox}[n]$

$$\hat{p}_{ox}[n] = \sum_{i,j} P_F[n; i, j] \tag{4.1}$$

Let $\vec{h}$ or $\mathbf{h}$ represent an L+1-Tap FIR bandpass filter. In our experiments, a 20-tap filter with a 14dB stopband suppression and passband cutoffs at 0.5Hz and 5Hz yielded good results. In case of pulse oximeter, it is shown in Kumar *et al.* [2019] that envelope normalization gives the best result. Hence, a low pass filter of similar taps as the bandpass filter was constructed with stopband edge at 0.5Hz. Let this be represented by $\vec{h_l}$ or $\mathbf{h}_l$. The filtered pulse oximeter reference waveform is given as $\tilde{p}_{ox}[n]$

$$\tilde{p}_{ox}[n] = \frac{\sum_{m=0}^{m=L} h(m)\hat{p}_{ox}[n-m]}{\sum_{m=0}^{m=L} h_l(m)\hat{p}_{ox}[n-m]} \tag{4.2}$$

The skin-ROI images are stacked up together along time in the form of a 3D-Tensor. Similarly, these are also filtered temporally for each pixel. The filtered skin

images is denoted by $\tilde{P}_S[n;i,j]$.

$$\tilde{P}_S[n;i,j] = \frac{\sum_{m=0}^{m=L} h(m)P_S[n-m;i,j]}{\sqrt{\sum_{m=0}^{m=L}(h(m))^2}} \tag{4.3}$$

Let C represent the number of cycles of the sampling rate over which we would like to take an inner product along time between $\tilde{p}_{ox}$ and $\tilde{P}_S$ in order to estimate the blood perfusion. In our experiments, 5 heart cycles or $C = 5$ or inner product over 100-150 frames provided a good results. Inner product over this region basically averages out noise in our measurements. The blood perfusion estimate $\hat{A}[n;i,j]$ over the skin-ROI is given as

$$\hat{A}[n;i,j] = \sum_{c=0}^{c=C} \tilde{P}_S[n-c;i,j]\tilde{p}_{ox}[n-c] \tag{4.4}$$

4.4 Implementation

The above section gave an overview of the mathematical algorithms which need to be implemented. This section will dive deep into the implementation details.

4.4.1 Platform

The "Android" Developers [2011] platform was chosen to implement this project. This is because of the widespread developer support tools in the form of Android Studio Studio [2017]. Also, Android phones are widely used in a developing country like India. Since Google provides very good camera APIs bundled with the Android platform, this seemed like the best way to go forward.

4.4.2 App Architecture

The application can be split into two main components: a user interface part (frontend) visible to the user and an information processing part (backend) invisible to the user. The frontend is coded completely in Java and Kotlin (as advised by Google) and the backend is a mix of C++ and Java. The frontend and the backend are coded in such a way, that the backend could even be ported out of Android and run on any other machine with minimal additional interfacing code. In order to run highly-powerful and parallelized computer vision algorithms, we resorted to OpenCV. OpenCV Bradski and Kaehler [2008] is one of the most powerful computer vison libraries out there. It comes preloaded with a host of useful computer vision algorithms. However, a suitable version of OpenCV for our specifications wasn't available out-of-box. Hence we had to compile our own ad-hoc OpenCV library for the ARMv7-abi generic Android platform.

4.4.3 Synchronizing Cameras

Accessing dual cameras at the same time in a smartphone is on its own a herculean task. Basically, camera pipelines tend to send data using dedicated video digital signal processors. Some OEMs tend to lock down on multiple streams of data running through these DSPs. Hence, some phones don't allow two camera streams to be opened simultaneously. However, with a recent Camera2 API update, Google has taken down this menace and made it possible to access multiple cameras.

But there is a catch. The camera streams are not synchronized in time. Nor does the sampling happen exactly at the desired frame rate i.e 20 or 30 Hz. To solve this, we used APIs to get the camera capture time in nanosecond resolution.

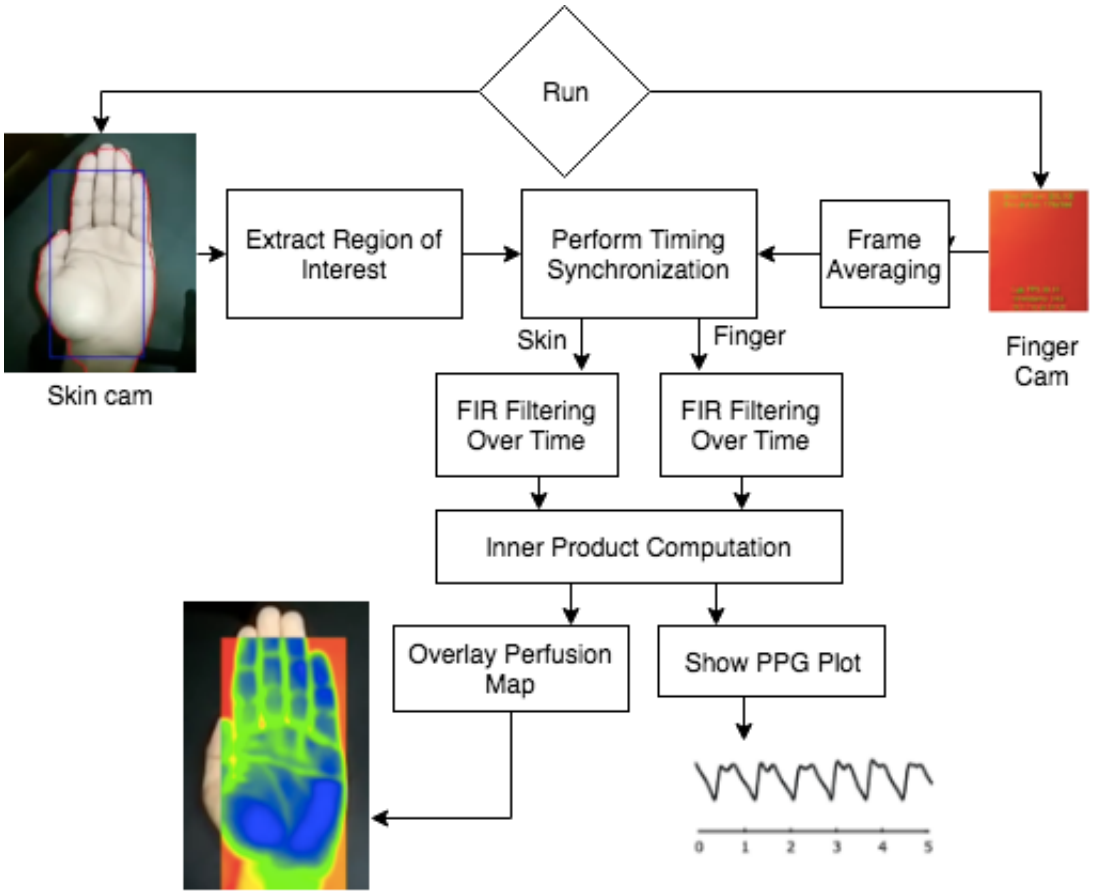


Figure 4.3: Mobile Pulsecam Implementation Pipeline

Based on this, we linearly interpolated through the images in time to obtain equally spaced sampling points. This was fed to the blood perfusion estimating algorithm. Motion changes are taken care of through a MOSSE Bolme *et al.* [2010] tracker for the skin region of interest.

4.4.4 Displaying Outputs

In order to display the outputs, a RAINBOW colormap is used. In order to be able to view all the regions, an adaptive normalization was performed on the input region. Because of this, we can't exactly get an absolute value of perfusion. Rather, we get a relative overview over the region of interest.

A schematic of the implementation pipeline is shown in Figure 4.3.

CHAPTER 5

Results

The result analysis of Mobile Pulsecam can be split into two folds

- First, analyze Pulsecam Kumar *et al.* [2016] and benchmark its effectiveness.
- Second, analyze Mobile Pulsecam in comparison with Pulsecam.

5.1 Analyzing Pulsecam

The current defacto medical standard for producing blood perfusion maps is Laser Doppler and Speckle imaging. So, by default, in order to make a confident case for Pulsecam, it has to be compared with these. The best way to do this is through clinical study.

In their initial Pulsecam Kumar *et al.* [2016] paper, Rice had performed a post occlusive reactive hyperemia (POHR) test on 4 healthy individuals to measure the change in blood perfusion in their palm before and after an occlusion event. More information about POHR tests can be found in Morales *et al.* [2005]. Occlusion of blood flowing to the palm is done using a standard pressure cuff put on the arm of the same hand. Blood volume waveform is recorded using a pulse oximeter from the middle finger of the other hand. On an average, they see an SNR improvement of 0.5-3 dB per pixel block in the blood perfusion map derived from Pulsecam compared to camera-only methods. Further, it was observed that the algorithm worked well for several skin tones too. These showed that Pulsecam could be touted as a winner among the alternative blood perfusion imaging space.

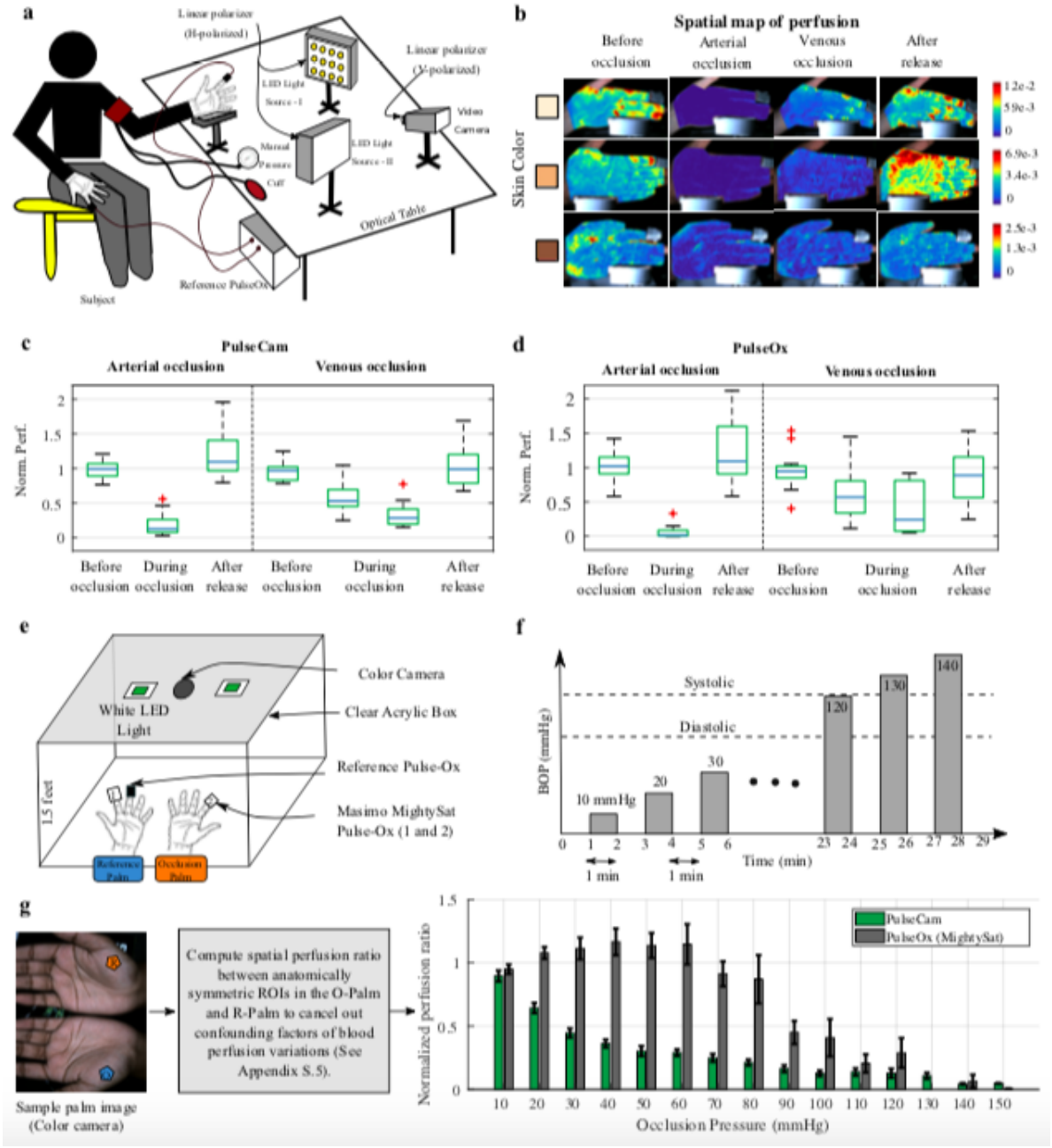


Figure 5.1: Clinical study conducted by Rice for Pulsecam Kumar *et al.* [2019]

In their upcoming Nature paper Kumar *et al.* [2019], Rice has performed extensive clinical testing to compare head-on between Laser Doppler imaging and Pulsecam. A example setup is shown in Figure 5.1. They collected datasets from live surgeries at operation theatres. They have done a very expansive data analysis. Through this, they claim the optimality of Pulsecam.

5.2 Analyzing Mobile Pulsecam

Strictly speaking, a clinical study would be the best way to test Mobile Pulsecam. However, due to time and resource constraints, we resorted to the quicker way. Pulsecam has already been shown to be convincingly optimal. So we obtain a video sample, run it through Pulsecam and through Mobile Pulsecam and then compare the outputs.

The experimental setup is shown in Figure 5.1(a). On one arm of the person, there is an occlusion cuff which is slowly released during video capture. As a result, initially the perfusion should be less. However, as time progresses, the perfusion will increase. The output of Pulsecam is shown in 5.2. In this particular color map (Figure 5.2(c)), the more redder the more perfusion it is. The output of Mobile Pulsecam is shown in Figure 5.3. In this color map, the more bluer the output, the more perfusion present. By analyzing qualitatively, the perfusion at each place in the region of interest in Figure 5.2 and Figure 5.3 should increase from occlusion cuff closing to opening. This is definitely observed. In Figure 5.3, when occlusion cuff is opened, some yellowish areas become greener, which means perfusion improves here. The general trends follow. One drawback though from Mobile Pulsecam is that we don't get highly accurate absolute perfusion estimates

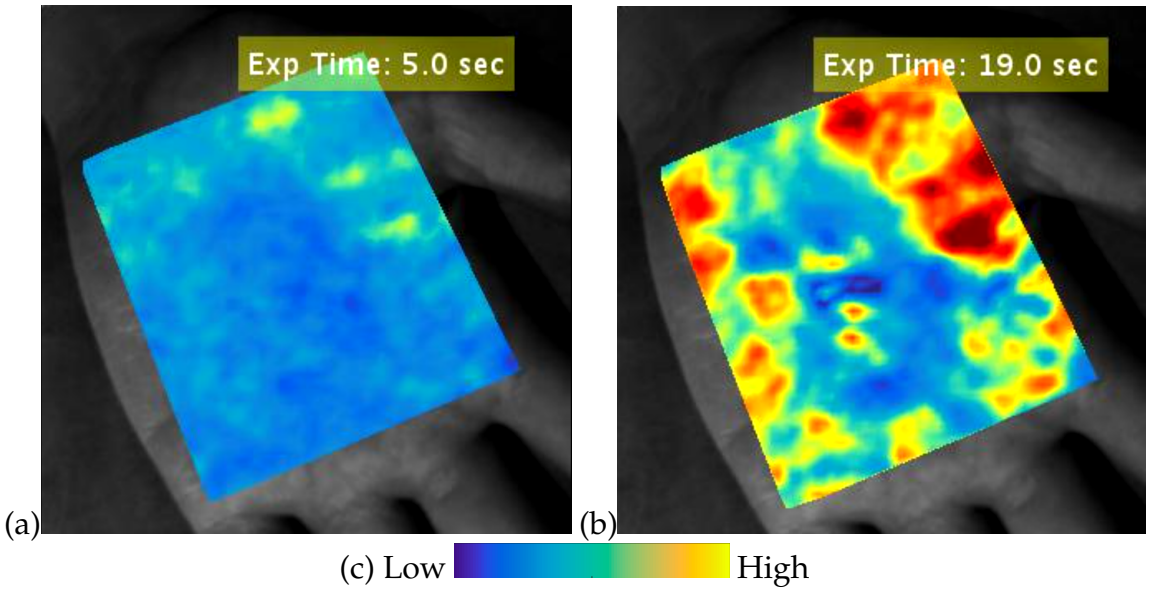


Figure 5.2: Output of Pulsecam Kumar *et al.* [2019]: Occlusion cuff (a) closed (b) opened (c) colormap

in various parts of the region of interest. This is because of scaling to display the output and color mapping.

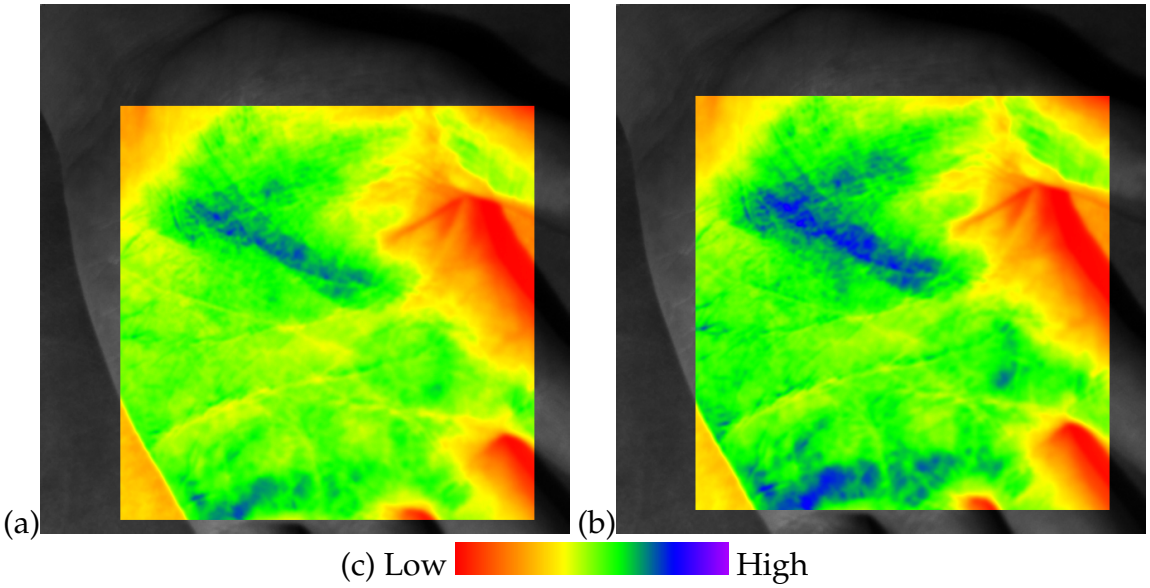


Figure 5.3: Output of Mobile Pulsecam: Occlusion cuff (a) closed (b) opened (c) colormap

CHAPTER 6

Conclusion

Pulsecam Kumar *et al.* [2016] is a revolutionary invention by Rice University in providing scalable, cheap, ubiquitous healthcare for all. Pulsecam aims to be a low-cost alternative to Laser Doppler and Speckle Imaging through using just a camera and a pulse oximeter. One of the main applications of these bulky, costly doppler imaging tools is blood perfusion imaging. Through Pulsecam Kumar *et al.* [2019], Rice show that they can obtain clinical level results just using a low-cost camera and pulse oximeter. Through cheap hardware and high-quality software, RIce is able to challenge conventional blood perfusion imaging techniques.

In this project, we built on Mobile Pulsecam, a blood perfusion technique which is capable of running on your smartphone. We further replace the pulse oximeter in Rice's Pulsecam by a secondary camera. We realized that this means any pair of two cameras should be able to do the job. This made us inclined towards building an algorithm which would work on dual-camera smartphones. In order to make a computationally intensive algorithm run on hand-held hardware, we modified and tweaked Rice's formulation for the smartphone without compromising much on its performance. All these were detailed in the previous sections. The screenshots of various sections in the app, based on our architecture, is shown in Figure 6.1.

The Mobile Pulsecam algorithm is capable of working in real-time on commonly available dual camera smartphones. The user places an illuminated finger over one camera (which acts as a pulse oximeter) and shows the skin region of interest to the other camera. Through sensor fusion, we are able to construct high-quality blood

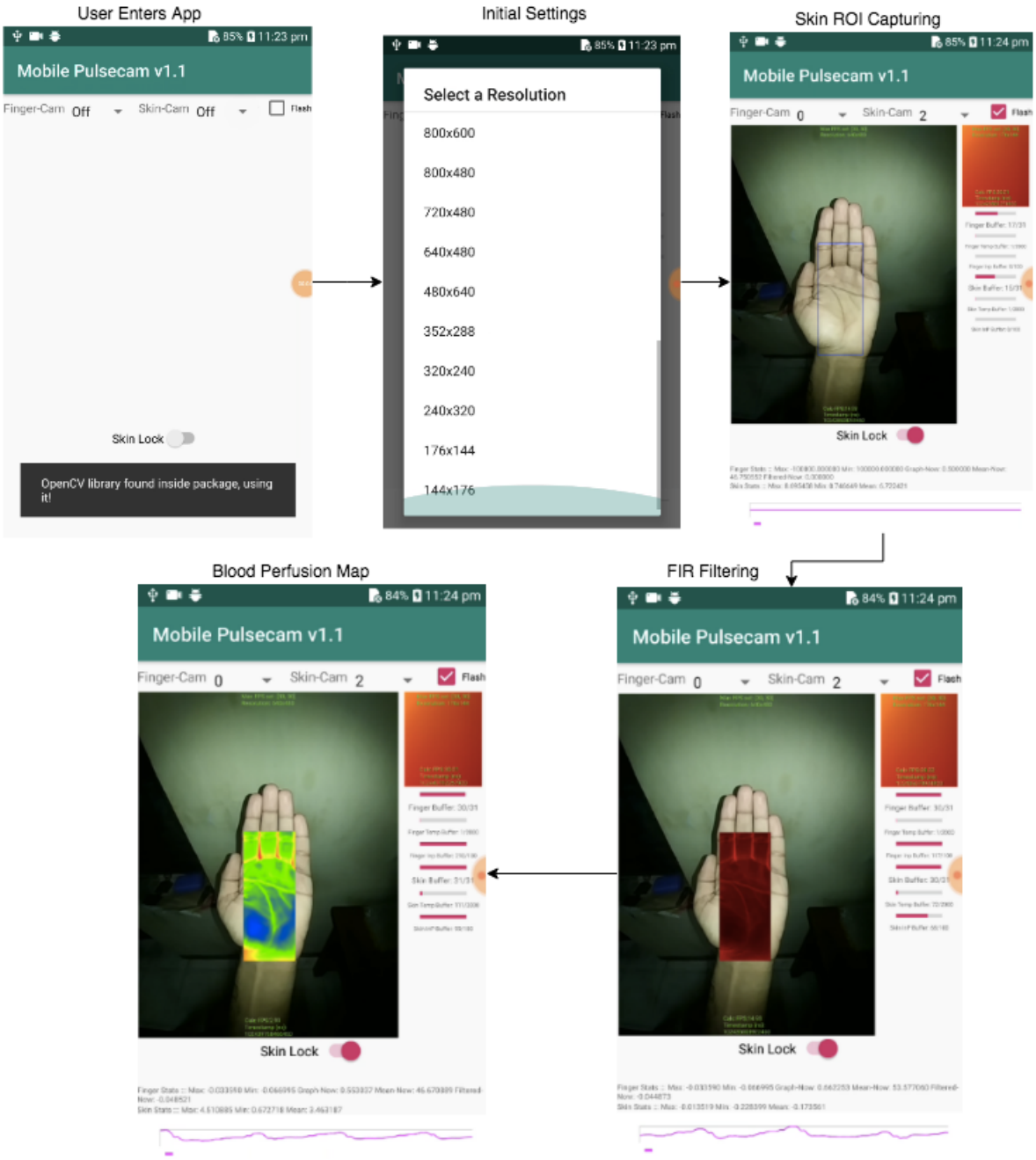


Figure 6.1: Palette Showcasing Different Parts of the Mobile Pulsecam App

perfusion maps. Though a major drawback of this is that absolute magnitude of perfusion estimate can't be recovered. This would be a good research problem to work in the next iteration of Mobile Pulsecam.

We strongly believe that one day, the quality of smartphone health diagnostics like Mobile Pulsecam would improve to such a good extent that people can get high-quality testing done right from the comfort of their homes. This would send ripples across the medical industry, making preventive testing/diagnostics accessible to all at almost no cost. We are very optimistic about such a situation being just around the corner.

REFERENCES

Allen, J. and **K. Howell** (2014). Microvascular imaging: techniques and opportunities for clinical physiological measurements. *Physiological measurement*, **35**(7), R91.

Binnendijk, E., **R. Koren**, and **D. M. Dror** (2012). Hardship financing of healthcare among rural poor in orissa, india. *BMC health services research*, **12**(1), 23.

Bolme, D. S., **J. R. Beveridge**, **B. A. Draper**, and **Y. M. Lui**, Visual object tracking using adaptive correlation filters. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010.

Bradski, G. and **A. Kaehler**, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

Corstian, A., **E. Klijn**, **W. K. Lagrand**, **J. J. Brugts**, **C. Ince**, **P. E. Spronk**, and **M. L. Simoons** (2008). The microcirculation in health and critical disease. *Progress in cardiovascular diseases*, **51**(2), 161–170.

Developers, A. (2011). What is android.

Forrester, K. R., **J. Tulip**, **C. Leonard**, **C. Stewart**, and **R. C. Bray** (2004). A laser speckle imaging technique for measuring tissue perfusion. *IEEE transactions on biomedical engineering*, **51**(11), 2074–2084.

Gangolli, L. V., **R. Duggal**, **A. Shukla**, *et al.*, *Review of healthcare in India*. Centre for enquiry into health and allied themes Mumbai, 2005.

Gregoski, M., **A. Vertegel**, and **F. Treiber**, Photoplethysmograph (ppg) derived heart rate (hr) acquisition using an android smart phone. *In Proceedings of the 2nd Conference on Wireless Health*. ACM, 2011.

Kamshilin, A. A., **S. Miridonov**, **V. Teplov**, **R. Saarenheimo**, and **E. Nippolainen** (2011). Photoplethysmographic imaging of high spatial resolution. *Biomedical optics express*, **2**(4), 996–1006.

Kamshilin, A. A., **V. Teplov**, **E. Nippolainen**, **S. Miridonov**, and **R. Giniatullin** (2013). Variability of microcirculation detected by blood pulsation imaging. *PloS one*, **8**(2), e57117.

Kumar, M., **J. Suliburk**, **A. Veeraraghavan**, and **A. Sabharwal**, Pulsecam: High-resolution blood perfusion imaging using a camera and a pulse oximeter. *In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016.

Kumar, M., **J. Suliburk**, **A. Veeraraghavan**, and **A. Sabharwal** (2019). Pulsecam: a non-contact motion-robust and highly sensitive blood perfusion imaging modality. *Submitted to Nature Reviews Health*.

Kumar, M., **A. Veeraraghavan**, and **A. Sabharwal** (2015). Distanceppg: Robust non-contact vital signs monitoring using a camera. *Biomedical optics express*, **6**(5), 1565–1588.

McClellan, J. H. and **T. W. Parks** (2005). A personal history of the parks-mcclellan algorithm. *IEEE Signal Processing Magazine*, **22**(2), 82–86. ISSN 1053-5888.

Morales, F., **R. Graaff**, **A. J. Smit**, **S. Bertuglia**, **A. L. Petoukhova**, **W. Steenbergen**, **P. Leger**, and **G. Rakhorst** (2005). How to assess post-occlusive reactive hyperaemia by means of laser doppler perfusion monitoring: application of a standardised protocol to patients with peripheral arterial obstructive disease. *Microvascular research*, **69**(1-2), 17–23.

Rao, M. and **D. Mant** (2012). Strengthening primary healthcare in india: white paper on opportunities for partnership. *Bmj*, **344**, e3151.

Rubins, U., **V. Upmalis**, **O. Rubenis**, **D. Jakovels**, and **J. Spigulis**, Real-time photoplethysmography imaging system. *In 15th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics (NBC 2011)*. Springer, 2011.

Serov, A., **B. Steinacher**, and **T. Lasser** (2005). Full-field laser doppler perfusion imaging and monitoring with an intelligent cmos camera. *Optics express*, **13**(10), 3681–3689.

Studio, A. (2017). Android studio. *The Official IDE for Android*.

Wardell, K., **A. Jakobsson**, and **G. E. Nilsson** (1993). Laser doppler perfusion imaging by dynamic light scattering. *IEEE Transactions on biomedical Engineering*, **40**(4), 309–316.