

# VIDEO-BASED STANDOFF HEALTH MEASUREMENTS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jeehyun Choe

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF DISSERTATION APPROVAL**

Dr. Edward J. Delp, Chair

School of Electrical and Computer Engineering

Dr. A.J. Schwichtenberg

Department of Human Development and Family Studies

Dr. Mary L. Comer

School of Electrical and Computer Engineering

Dr. Yung-Hsiang Lu

School of Electrical and Computer Engineering

**Approved by:**

Dr. Dimitrios Peroulis

Head of the School of Electrical and Computer Engineering

## ACKNOWLEDGMENTS

Joining and going through Purdue PhD program has been a big challenge for me. I would like to express appreciation to those who has motivated, guided, helped, and inspired me on my PhD research.

First, I would like to thank my advisor, Prof. Edward J. Delp for giving me opportunity to study in Video and Image Processing Laboratory (VIPER). Having his guide and support on my research was invaluable. I especially appreciate his support during my difficult times. I learnt and benefited a lot from the lab working environments that Prof. Delp has created and maintained. Listening and reading interesting tech and non-tech stories that he has constantly shared to VIPER were another fun part of being a lab member.

I would like to thank Prof. A.J. Schwichtenberg for her active role and engagement on my research projects. She shared numerous advices and comments on my research through weekly meetings and put lots of time on paper revisions. It was exciting to do research work on real-world problems that can impact many families. I appreciate Prof. A.J. and her lab for letting me work on the real-world data and guiding me focus on important problems.

I would like to thank my other committee members, Prof. Mary Comer and Prof. Yung-Hsiang Lu for their time and valuable comments. I appreciate Prof. Pizlo's insightful comments during my preliminary exam.

I would like to thank my VIPER colleagues, who have given their support: Daniel, Dahjung, Bee, Yuhao, Javi, Shaobo, Soonam, David G, Cici, Neeraj, Khalid, Joon-soo, He Li, Yu Wang, Chichen, Shuo, Sri, Blanca, Albert, Ye, Ruiting, David Ho, Di, Chang, Bin, Joy, Zeman, Sriram, Emily, and Hans. Being surrounded by people who seek for similar values (passion in research, pursuing contributions to the community as professionals) was a great motivation to me. I would also like to thank

Developmental Studies Laboratory members for their support: Pearlynne, Ashleigh M Kellerman, and Emily Abel.

I would like to thank Ilia Potuzhnov for his engagement on designing and implementing the Sleep Web App. His advice and comments on user experiences and continuous guide on implementation was instrumental for this Web App development. His guidance on setting up nice working environments with automatic testing scripts made the work flow so much easier. I also appreciate his support for encouraging me to finish up PhD thesis while I work at Google.

I would like to thank my mother, Hye-Sook Kim, father, Jae-Woong Choe, and brother, Duk-Hyun Choe, for their endless love and support. It was sad to see my two grandfathers, Chang-Souk Kim and Man-Ha Choe, passing away during my PhD. I appreciate Man-Ha Choe for leaving me his hand-written letter before he passed away—Grandpa, as you have written, I am finishing up my PhD. I thank all my families for their love and support.

I appreciate Purdue University for providing me many years of financial supports through Graduate TA and Graduate Instructor positions. My long journey as a PhD student did not always turned out as I wanted but I gained valuable lessons and learnt how to deal with difficulties of many kinds. I really enjoyed spending time on reading, learning, experimenting, thinking and discussing about interesting research problems with others. Purdue has been a great place for that.

This thesis used video and image data collected in different research groups at Purdue University. The human videos used for experiments in Chapter 2 and 3 were obtained in cooperation with Video and Image Processing Laboratory (VIPER) and Developmental Studies Laboratory in Purdue University. The infant sleep videos used in Chapter 4, 5 and Appendix A were provided from Developmental Studies Laboratory, Purdue University. The IP addresses of web cameras in Chapter 6 were selectively obtained from the camera database in High-Efficiency, Low-Power System Group, Purdue University.

# TABLE OF CONTENTS

|  | Page |
|--|------|
| LIST OF TABLES . . . . .   | viii |
| LIST OF FIGURES . . . . .  | ix   |
| ABBREVIATIONS . . . . .  | xii  |
| ABSTRACT . . . . .   | xiv  |
| 1 INTRODUCTION . . . . .   | 1    |
| 1.1 Video-Based Standoff Health Measurements . . . . .                       | 1    |
| 1.1.1 Heart Rate (HR) Measurements . . . . .                                 | 3    |
| 1.1.2 Sleep Analysis . . . . .   | 14   |
| 1.2 Image-Based Geographical Location Estimation Using Web Cameras . . . . . | 17   |
| 1.3 Contributions of This Thesis . . . . .                                   | 20   |
| 1.4 Publications Resulting From This Thesis . . . . .                        | 22   |
| 2 PROPOSED APPROACH FOR<br>RESTING HEART RATE ESTIMATION . . . . .           | 24   |
| 2.1 Overview of Proposed System . . . . .                                    | 24   |
| 2.2 Frequency Clusters . . . . .   | 27   |
| 2.3 AFR Estimation by Background Removal . . . . .                           | 28   |
| 2.4 Face Tracking and Skin Detection . . . . .                               | 29   |
| 2.5 Experimental Results . . . . .   | 32   |
| 3 UNDERSTANDING MOTION EFFECTS<br>IN VIDEOPLETHYSMOGRAPHY (VHR) . . . . .    | 42   |
| 3.1 Motion and Illumination in VHR . . . . .                                 | 42   |
| 3.2 Simple Modeling: Intensity Change of Moving Object . . . . .             | 44   |
| 3.3 Intensity Model in Human Video with Motion . . . . .                     | 51   |
| 3.4 Filters . . . . .  | 53   |

|   | Page |
|---|------|
| 3.5 Region Selection and Face Direction . . . . .                             | 54   |
| 3.6 Conclusion and Future Work . . . . .                                      | 56   |
| 4 SLEEP ANALYSIS USING<br>MOTION AND HEAD DETECTION . . . . .                 | 62   |
| 4.1 Sleep Detection . . . . .   | 63   |
| 4.1.1 Motion Detection . . . . .  | 63   |
| 4.1.2 Reference Size Using Head Detection . . . . .                           | 65   |
| 4.1.3 Sleep Scoring . . . . .   | 66   |
| 4.2 Experimental Results . . . . .  | 69   |
| 4.3 Conclusions . . . . .   | 70   |
| 5 CLASSIFICATION OF SLEEP VIDEOS<br>USING DEEP LEARNING . . . . .             | 72   |
| 5.1 Introduction . . . . .  | 72   |
| 5.2 Related Work . . . . .  | 73   |
| 5.2.1 Motion and Long Term Dependencies in VSG . . . . .                      | 73   |
| 5.2.2 Long Short- Term Memory Networks (LSTM) . . . . .                       | 74   |
| 5.2.3 Video Classification Using Deep Learning . . . . .                      | 74   |
| 5.3 Proposed Method . . . . .   | 76   |
| 5.3.1 Motion Detection/Motion Index . . . . .                                 | 77   |
| 5.3.2 Loss Function . . . . .   | 78   |
| 5.4 Experiments . . . . .   | 78   |
| 5.4.1 Dataset . . . . .   | 78   |
| 5.4.2 Implementation Details . . . . .  | 80   |
| 5.4.3 Results . . . . .   | 80   |
| 5.5 Conclusions . . . . .   | 84   |
| 6 IMAGE-BASED GEOGRAPHICAL LOCATION<br>ESTIMATION USING WEB CAMERAS . . . . . | 86   |
| 6.1 Sunrise/Sunset Estimation . . . . .                                       | 86   |
| 6.2 Sky Region Detection . . . . .  | 88   |

|  | Page |
|--|------|
| 6.3 Estimating Location from Sunrise/Sunset . . . . .          | 89   |
| 6.4 Experimental Results . . . . .                             | 91   |
| 7 CONCLUSION . . . . .   | 96   |
| 7.1 Summary . . . . .  | 96   |
| 7.2 Future Work . . . . .                                      | 98   |
| 7.3 Publications Resulting From This Thesis . . . . .          | 99   |
| REFERENCES . . . . .   | 101  |
| A SLEEP WEB APPLICATION . . . . .                              | 115  |
| A.1 Introduction . . . . .                                     | 115  |
| A.2 Sleep/Awake Classification in Sleep Web App . . . . .      | 116  |
| A.3 User Manual . . . . .                                      | 117  |
| A.3.1 File Preparation . . . . .                               | 117  |
| A.3.2 File Uploading . . . . .                                 | 120  |
| A.3.3 Results . . . . .  | 122  |
| A.4 Environments . . . . .                                     | 124  |
| A.4.1 Installations (system level) . . . . .                   | 124  |
| A.4.2 Installations (for sleep/awake classification) . . . . . | 126  |
| B SOURCE CODE . . . . .  | 127  |

## LIST OF TABLES

| Table   | Page |
|---|------|
| 2.1 Number of non-empty bins for $32^3$ color bins in UCSB dataset. . . . .   | 32   |
| 2.2 A Comparison of Two Methods for Dataset 1 . . . . .   | 39   |
| 2.3 A Comparison of Two Methods for Dataset 2, No-motion videos. . . . .  | 40   |
| 2.4 A Comparison of Two Methods for Dataset 2, Non-random motion videos. . . . .                                      | 41   |
| 3.1 $r_L = L_{min}/L_{max}$ when $ 2d  \leq 11/12$ , $D = 11$ , and $r = 7/12$ . . . . .                              | 48   |
| 3.2 $r_L = L_{min}/L_{max}$ [%]. . . . .  | 50   |
| 4.1 Auto-VSG ( $c = 5$ ) vs. B-VSG Labeling. . . . .  | 68   |
| 4.2 Auto-VSG ( $c = 1$ ) vs. Actigraphy. . . . .  | 68   |
| 5.1 Training/Test Set Division of Sleep Dataset. . . . .  | 79   |
| 5.2 Results [%] for the number of test GoPs $n = 96,015$ . Models trained using<br>loss with uniform weights. . . . . | 81   |
| 5.3 Results [%] for the number of test GoPs $n = 96,015$ . Models trained using<br>weighted loss. . . . .             | 82   |
| 6.1 Sunrise/sunset detection for camera01 for using $th_{mean}$ . . . . .   | 93   |
| 6.2 The result for latitude for using $th_{mean}$ . . . . .   | 93   |
| 6.3 The result for longitude for using $th_{mean}$ . . . . .  | 94   |



## LIST OF FIGURES

| Figure  | Page |
|---|------|
| 1.1 Examples of Heart Rate (HR) Measurement Settings: (a) Finger Pulse Oximeter; The sensor is attached to the finger (b) Video-based method. . .   | 2    |
| 1.2 Examples of VSG Settings: (a) Traditional method; The sensor is attached to the ankle (b) Video-based method. . . . .   | 2    |
| 2.1 The block diagram of the proposed system (after [28, 29]). . . . .  | 25   |
| 2.2 An example of frequency clusters. $P_{max}$ is the maximum value of the PSD within $[f_l, f_h]$ . $t_r$ is a parameter used to determine the weak signals as described in Section 2.2. $t_n$ is a parameter used to determine the neighboring clusters as described in Section 2.2. If two clusters formed by thresholding $P[k]$ are with $t_n$ Hz of one another we considered them ‘neighbors’ and merge them into one cluster. Cluster 1 and Cluster 2 in this Figure are not ‘neighbors’ because $ f_{2_h} - f_{1_l}  > t_n$ . $N$ is the number of points in the positive frequency domain, and $k$ is the index in the frequency domain, $f_s$ is the sampling rate. . . . . | 27   |
| 2.3 An example of matching clusters from the face signal (top) and the background signal (bottom). $P[k]$ is the PSD of the signals and $k$ is the index in the frequency domain. . . . .   | 29   |
| 2.4 Comparison of two different quantizations on skin pixels. . . . .   | 31   |
| 2.5 The block diagram of the proposed skin detection system. . . . .  | 32   |
| 2.6 Data Collection Environment. . . . .  | 34   |
| 2.7 Examples of Video Settings. . . . .   | 34   |
| 2.8 AFR obtained by the Proposed method for Dataset 1. . . . .  | 36   |
| 2.9 Estimated HRs and Ground Truth HR for Test 18 in Dataset 1. . . . .   | 37   |
| 2.10 AFR obtained by the Proposed method for Dataset 2, No-motion videos. .   | 37   |
| 2.11 AFR obtained by the Proposed method for Dataset 2, Motion videos. . . .  | 38   |

| Figure   | Page |
|--|------|
| 3.1 Average green trace within face skin region for 10-second duration for subject 17, Dataset 1. The range of intensity $L$ for all color channels in Dataset 1 is: $L \in [0, 255]$ . The average HR obtained from pulse oximeter for this 10-second duration is 64 bpm (meaning about 10.7 beats for 10 second). $L_{min} = 67.4$ , $L_{max} = 68.3$ and $L_{min}/L_{max} = 98.7\%$ . . . . .   | 45   |
| 3.2 Point analysis for a simple motion model. $\theta$ is the incident angle, $r$ is the radius of the head when viewed from the top, $d$ is the moved distance, $D$ is the distance between the source light and the line of movement, $\alpha$ is the angle from the head direction to the line connecting center of head and the light, $\beta$ is the angle between specific face point and head direction from the center of the head, and $\gamma$ is the angle between specific face point and the center of head from the light source. $\alpha$ is zero when the head direction is toward the light and aligned with the line between the source light and the center of the head. $\alpha$ is positive when in counterclock-wise direction. $\gamma$ is zero when the face point is on the line between the light source and the center of the head. $\gamma$ is positive when in counterclock-wise direction. In both figure(a) and (b), the leftmost circle denotes the farthest position to the left and the rightmost circle denotes the farthest position to the right. | 46   |
| 3.3 Relation between moved distance $d$ and $\cos\theta$ for various $\beta > 0$ . . . . .   | 47   |
| 3.4 The data collection environment. The distance $D$ between the object's moving plane and the light is 11 ft. The range of moving distance $d$ along the moving plane is $ 2d  < 11$ inch. The height of the light $h_l$ and height of the object $h_o$ are similar ( $h_l = 47$ and $h_o = 43$ inches). The object surface facing the camera is a paper in solid color of light pink. . . . .   | 48   |
| 3.5 Camera views of test videos in different angles. The average $L(n)$ is obtained from the ROI pixels within the green circle—the radius of the circle is the diagonal distance between two red points divided by 6.5. Four corner points in red are manually selected in the first frame and obtained by feature tracker [134] in the rest of the frames. . . . .   | 49   |
| 3.6 Average $L$ of ROI in R channel in three different surface angles: No-motion vs. Motion. $L$ is 8bits/pixel/channel and $L \in [0, 255]$ . The PSD of each trace (the average $L(n)$ ) within the frequency range of our interest in VHR, $f_l = 0.7$ and $f_h = 3.0$ Hz, is plotted in blue below the each trace.   | 59   |
| 3.7 Block Diagram. . . . .   | 60   |
| 3.8 An example captured from Dataset 2. Facial points denoted in red. ROI regions denoted in green—only the ROI in the middle of the nose was used.  | 60   |

| Figure   | Page |
|--|------|
| 3.9 Experimental result on Dataset 2 of non-random motion videos: Average $L(n)$ and $\hat{L}(n)$ . $\hat{L}(n)$ is denoted as “Estimated L” in the plot label. The red patches in PSD plots denote the GTHR range for each subject. The frequency range in PSD plot is $f_l = 0.7$ and $f_h = 2.0$ Hz. Both subject 3 and 14 showed strong peak around 0.17 Hz corresponding to motion (Not shown on the plot). . . . . | 61   |
| 4.1 Proposed Sleep Detection System. . . . .   | 63   |
| 4.2 Example of motion detection: Preprocessed image (left), background model (middle), and moved pixels denoted in white (right). . . . .  | 65   |
| 4.3 Examples of head detections of two different infants. . . . .  | 67   |
| 5.1 LRCN [168, 169]. GoP is a Group of Pictures. . . . .   | 75   |
| 5.2 C3D [170]. The C3D convolution kernel includes temporal depth in addition to 2-dimensional CNN kernel of width and height. . . . .   | 76   |
| 5.3 Proposed Sleep Detection System: Sleep/Awake Using a Motion Index and LSTM. . . . .  | 77   |
| 5.4 ROCs. Models trained using loss with uniform weights. GoP is Group of Pictures. . . . .  | 83   |
| 5.5 ROC curve. Models trained using weighted loss. GoP is Group of Pictures. . . . .   | 84   |
| 6.1 A collection of pairs of test images and their skymask. . . . .  | 90   |
| 6.2 The mean luminance of the entire image vs. the sky region. . . . .   | 92   |
| A.1 Block diagram for Sleep/Awake classification in Sleep Web App. . . . .   | 116  |
| A.2 Sleep Web App Manual: File Compressing. . . . .  | 118  |
| A.3 Sleep Web App Manual: File Compressing. . . . .  | 120  |
| A.4 Sleep Web App Manual: File Upload. . . . .   | 121  |
| A.5 Sleep Web App Manual: File Upload. . . . .   | 122  |
| A.6 Sleep Web App Manual: File Upload. . . . .   | 122  |
| A.7 Sleep Web App Manual: Results. . . . .   | 123  |
| A.8 Sleep Web App Manual: Final result page. Download buttons for per-minute sleep analysis result and sleep summary results are provided. . . .   | 123  |

## ABBREVIATIONS

|                  |                                |
|------------------|--------------------------------|
| HR               | Heart Rate                     |
| PPG              | Photoplethysmography           |
| VHR              | Videoplethysmography           |
| VSG              | Videosomnography               |
| B-VSG            | Behavioral-Videosomnography    |
| LSTM             | Long Short-term Memory         |
| C3D              | 3D Convolutional Networks      |
| bpm              | Beats per minute               |
| ECG or EKG       | Electrocardiography            |
| LED              | Light-emitting diode           |
| Hb               | Hemoglobin                     |
| HbO <sub>2</sub> | Oxyhemoglobin                  |
| BCG              | Ballistocardiography           |
| BSS              | Blind Source Separation        |
| ICA              | Independent Component Analysis |
| PCA              | Principal Component Analysis   |
| LDA              | Linear Discriminant Analysis   |
| LE               | Laplacian Eigenmap             |
| ROI              | Region Of Interests            |
| DFT              | Discrete Fourier Transform     |
| MUSIC            | Multiple Signal Classification |
| SNR              | Signal to Noise Ratio          |
| ISP              | Image Signal Processing        |
| AWB              | Automatic White Balance        |

|                  |  |
|------------------|--|
| AGC              | Automatic Gain Control                     |
| AVI              | Audio Video Interleave                     |
| BPF              | bandpass filter                            |
| MA               | Motion Artifacts                           |
| IRB              | Institutional Review Board                 |
| CAM <sup>2</sup> | Continuous Analysis of Many CAMeras system |
| GoP              | Group of Pictures                          |
| CNN              | Convolutional Neural Network               |
| RNN              | Recurrent Neural Network                   |
| VGE              | Vanishing Gradient Effect                  |
| LRCN             | Long-term Recurrent Convolutional Networks |
| ROC              | Receiver Operating Characteristic          |
| AUC              | Area Under the ROC Curve                   |
| ACC              | Accuracy                                   |
| PRE              | Precision                                  |
| REC              | Recall                                     |
| SPEC             | Specificity                                |

## ABSTRACT

Choe, Jeehyun Ph.D., Purdue University, August 2019. Video-Based Standoff Health Measurements. Major Professor: Edward J. Delp.

We addressed two interesting video-based health measurements. First is video-based Heart Rate (HR) estimation, known as video-based Photoplethysmography (PPG) or videoplethysmography (VHR). We adapted an existing video-based HR estimation method to produce more robust and accurate results. Specifically, we removed periodic signals from the recording environment by identifying (and removing) frequency clusters that are present in the face region and background. This adaptive passband filter generated more accurate HR estimates and allowed other applied filters to work more effectively. Measuring HR at the presence of motions is one of the most challenging problems in recent VHR studies. We investigated and described the motion effects in VHR in terms of the angle change of the subjects skin surface in relation to the light source. Based on this understanding, we discussed the future work on how we can compensate for the motion artifacts. Another important health information addressed in this thesis is Videosomnography (VSG), a range of video-based methods used to record and assess sleep vs. wake states in humans. Traditional behavioral-VSG (B-VSG) labeling requires visual inspection of the video by a trained technician to determine whether a subject is asleep or awake. We proposed an automated VSG sleep detection system (auto-VSG) which employs motion analysis to determine sleep vs. wake states in young children. The analyses revealed that estimates generated from the proposed Long Short-term Memory (LSTM)-based method with long-term temporal dependency are suitable for automated sleep or awake labeling.

# 1. INTRODUCTION

## 1.1 Video-Based Standoff Health Measurements

There has been growing interests and needs in frequent and continuous health monitoring. Commonly monitored health information includes Heart Rate (HR), Blood Pressure (BP), and Respiration Rate (RR). To measure these require dedicated equipment and special devices. With the need for in-home health monitoring or telemedicine, using camera sensors for health-monitoring has been in the limelight in various health measurements. One of the greatest advantages of using videos is that the measurement is convenient. Nowadays cameras are everywhere, and anyone can easily record videos. Another advantage of using camera is that unlike most of the medical sensors, video recording is not intrusive, and requires no contact to the body. While lots of important health information is contained in human videos, the information can be difficult to obtain because it is very labor intensive or impossible to be observed by human eyes. In this thesis, we address two interesting video-based health measurements and propose methods that use video processing, computer vision, and machine learning techniques to obtain health information hidden in the videos.

First is video-based Heart Rate (HR) estimation. One of the most important health information is monitoring the perfusion of the circulation as cardiopulmonary parameters such as blood pressure and blood flow [1]. Figure 1.1 illustrates commonly used HR measurement. Section 1.1.1 introduces the video-based HR estimation.

Another important health information addressed in this thesis is monitoring activities during sleep. Pediatric sleep medicine is a field that focuses on typical and atypical sleep patterns in children. Within this field, physicians, interventionist, and researchers record and label child sleep with particular attention to sleep onset time,

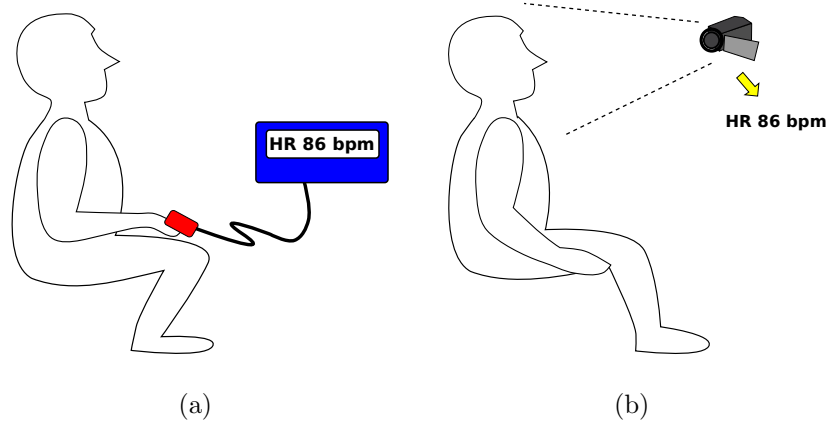


Fig. 1.1. Examples of Heart Rate (HR) Measurement Settings: (a) Finger Pulse Oximeter; The sensor is attached to the finger (b) Video-based method.

total sleep duration, and the presence or absence of night awakenings. One notable recording method is videosomnography (VSG) which includes the labeling of sleep/awake from video [2,3]. Traditional behavioral videosomnography (B-VSG) includes manual labeling of awake and sleep states by a trained technician/researcher [3]. Figure 1.2 shows a simple description of video-based VSG. Section 1.1.2 introduces the

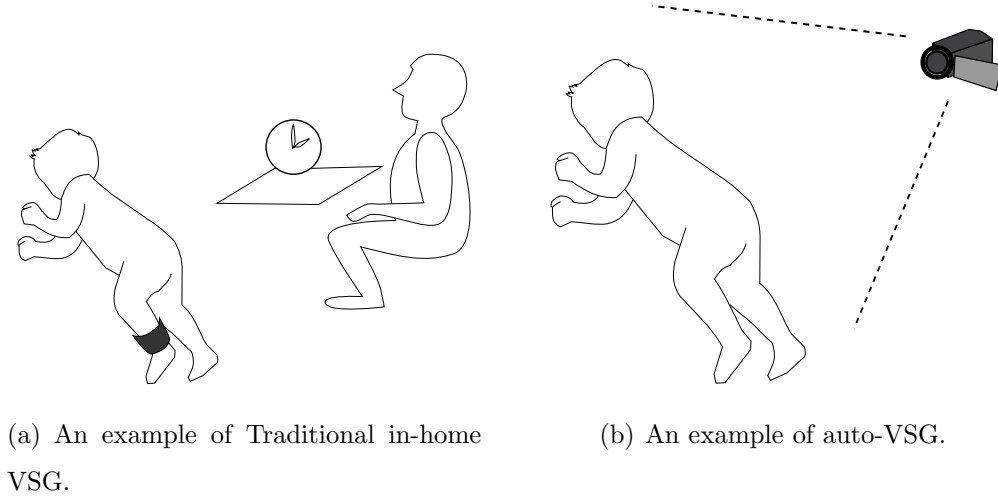


Fig. 1.2. Examples of VSG Settings: (a) Traditional method; The sensor is attached to the ankle (b) Video-based method.



video-based sleep analysis.

### 1.1.1 Heart Rate (HR) Measurements

Heart Rate (HR) is the number of heart beats or cardiac contractions per minute, often referred to as beats per minute (bpm). Normal HR varies from person to person such that children in younger age have faster HR and adults with higher fitness have slower HR (better cardiovascular fitness). The normal resting HR ranges for newborns 0 to 1 months of age is 70 to 190 bpm, infants 1 to 11 months is 80 to 160 bpm, children 1 to 9 years is 70 to 130 bpm, children 10 years and older and adults (including seniors) is 60 to 100 bpm and well-trained athletes is 40 to 60 bpm [4]. HR out of normal HR range can indicate health problems. Fast HR may signal an infection or dehydration [4]. A raise in your HR can also indicate stressed, anxious or “extraordinarily happy or sad” emotions. HR is routinely measured in clinical practice [5]. HR can be measured at areas where an artery passes close to the skin [4]. Current measurements of HR involve attaching devices/sensors to a patient’s fingers, arms, or chest. The two most commonly used techniques for measuring HR in clinical practice are Photoplethysmography (PPG) and electrocardiography (ECG or EKG).

The first HR measuring technique to introduce is Photoplethysmography (PPG). PPG is an optical technique that indexes blood volume changes in microvascular tissue to measure the rate of blood flow (or HR) [6]. The blood volume changes are represented as waveforms in PPG, called PPG waveform or PPG signal, and it is synchronized to each heart beat. The PPG waveform has been used since the 1930s [7,8]. It had been one of many methods for skin capillary blood flow measurement which include skin thermometry, thermal clearance, laser Doppler plethysmography, radioactive isotope clearance, electrical impedance methods [9]. In the 1980s, the pulse oximeter began to be used as routine clinical care and the importance of PPG wave-

form in clinical medicine greatly increased [7]. Using pulse oximeter is a traditional way of using PPG to measure HR.

The principle of traditional PPG is that when the light at a suitable part of the spectrum (near infrared) is directed into the skin, detection of the attenuated light which passes out of the skin gives a measure of its blood content where more blood present in the skin leads to greater attenuation of light [10]. The PPG waveform can be separated into an oscillating (ac) and a steady-state (dc) components and applications such as pulse counters, using the ac component, and skin color and hemoglobin saturation meters, using the dc component, are available [8, 9]. The peak-to-peak intervals obtained from ac component of PPG waveform represent heart cycles [11].

Elgendi [11] addressed that the quality of the PPG signal depends on the location and the properties of the subject's skin at measurement, including the individual skin structure, the blood oxygen saturation, blood flow rate, skin temperatures and the measuring environment. Challenges in obtaining PPG signal include poor contact between the body site and the photo sensor, variations in temperature, irregular heart beat caused by the premature ventricular beats (PVCs), and light interference from the measuring environment [11, 12].

Another technology used for monitoring HR is electrocardiography (ECG or EKG). ECG is a test that measures the electrical activity of the heart beat (i.e. the expansion and contraction of heart chambers) from an electrical impulse traveling through the heart [13]. ECG signals are acquired by placing Ag/AgCl electrodes on clearly defined anatomical positions and one lead (channel) of ECG recording requires three electrodes to produce the signal thus requiring three wires to be connected to the subject [14]. Clinical ECG recordings commonly use 3 to 12 leads [15], as opposed to PPG recording typically use only one probe. [15] suggested that PPG may prove a practical alternative to ECG for HR Variability (HRV) analysis since PPG provides accurate interpulse intervals from which HRV measures can be accurately derived in healthy subjects under ideal conditions.

Pulse oximeters use PPG to estimate HR [6]. In the early 1990s pulse oximetry became a mandated international standard for monitoring during anaesthesia [6]. Pulse oximeters are commonly used in clinical practice because of their low cost, high-accuracy, and relative ease of use. Pulse oximeters function on the principle that hemoglobin (Hb) and oxyhemoglobin (HbO<sub>2</sub>) absorb red and infrared light differently [12]. It measures the amount of red and infrared light that passes through the skin—as skin fills with blood or “blushes,” the ratio of red to infrared light changes. Commonly used body sites for placement of the pulse oximeter probe are fingers and earlobes but other sites such as toes, cheeks, nose, and tongue can be used as well [12].

Recently there are many wearable PPG sensors used for daily activities. Wearing PPG sensors on the fingers during daily activities is not well suited and different measurement sites have been explored extensively, including the ring finger, wrist, brachia, earlobe, external ear cartilage, the superior auricular region, forehead, and glasses-type system [1]. The wearable PPG has two different modes—transmission mode and reflectance mode based on the placement of light-emitting diode (LED) and photodetector (PD) [1]. Tamura *et al.* [1] addressed that while IR or near-IR wavelengths are better for measurement of deep-tissue blood flow, green LED has much greater absorptivity for both hemoglobin (Hb) and oxyhemoglobin (HbO<sub>2</sub>) compared to infrared light that green-wavelength PPG devices are becoming increasingly popular. Poh *et al.* [16] estimated HR using modified earphones with a regular cell phone. They obtained PPG signal through specially designed earphone where the earbuds are embedded with reflective photosensor. Health monitoring based on PPG method by using smartphone’s videocamera is addressed in several papers [17, 18]. Using a smartphone requires a finger to be placed on the smartphone’s camera in the way that it covers both the camera lens and the LED (the flash) [17, 18]. This can not be used during the activities but provides an easy access to HR measurement since it does not require any special equipment.

Other HR measurement methods include Ballistocardiography (BCG), a method for obtaining a representation of the heart beat-induced repetitive movements of the

human body, occurring due to acceleration of blood as it is ejected and moved in the large vessels [19]. BCG signal can be obtained by piezoelectric force sensors [19]. Paalasmaa *et al.* [20] estimated beat-to-beat HR from ballistocardiograms acquired with force sensors. Hernandez *et al.* [21] made use of a head-worn camera (Google Glass) that captures the view of the wearer to monitor subtle periodic BCG motions.

Bioimpedance measurements can be used to detect HR. Gonzalez-Landaeta *et al.* [22] obtained heart-related impedance changes when standing, by using four platform-type aluminum electrodes, and compared the bioimpedance signal to the ECG recordings.

All the methods explained above involve attaching devices/sensors to the human body. This can bring discomfort to many subjects/patients especially when they need to measure the vital signs frequently. Baby/patients with tactile sensitivities or patients over long periods of monitoring may not tolerate attaching sensors on their skin. Researchers have been investigating methods to overcome the drawback of HR estimation involving contacts with the body.

Millimeter-wave sensors together with color and depth cameras [23] have been used to estimate HR. They estimated HRs using a 94-GHz sensor to obtain the chest displacement corresponding to heartbeats and compared the result with ECG based HRs. Adib *et al.* [24] described a system called Vital-Radio that monitors HR of multiple people. They transmit a low-power wireless signal and obtain the time it takes for the signal to reflect back to the device where the wireless signals operate through walls. HR is estimated for each 10-second window and it captures the skin vibrations due to heartbeats which is BCG movements from the head, torso and buttock [24]. Their system detects periods during which the person is quasi-static and estimate HR only during such intervals since accurate HR estimation cannot be provided when the person walks or moves.

The video-based HR estimation, also known as videoplethysmography (VHR), mostly uses PPG methods and assess facial/skin region “micro-blushing.” Basic assumption in VHR methods is that small color variations, micro-blushing, in the

face/skin region reflect PPG signals (i.e., heart-beats). Remote, stand-off methods for assessing HR have emerged in the past years [25–79].

Several video-based approaches have been proposed for HR estimates using PPG. One of the approaches used from the early VHR is using the mean pixel values of green channel in the face region to obtain the PPG signal [26, 32, 34–36, 73]. Verkrusse *et al.* [26] addressed that while all RGB channel in a simple consumer level digital camera contained PPG information, the green channel featured the strongest PPG signal. Kwon *et al.* [34] experimented with a smartphone camera and reported that the green channel trace contains a relatively strong PPG signal more than other channels. Kumar *et al.* [35] explained that the green channel performs better because the absorption spectra of hemoglobin (Hb) and oxyhemoglobin (HbO<sub>2</sub>), two main constituent chromophores in blood, peaks in the region around 520–580 nm, which is essentially the passband range of the green filters in color cameras.

The PPG signal contained in the video is relatively small compared to various environmental factors including illumination change, camera-related signals and subjects’ movements. Blind Source Separation (BSS) is a technique for recovering a set of signals of which only “blindly” processed linear mixtures are observed [80]. Independent Component Analysis (ICA), one of BSS, is a technique for uncovering statistically independent source signals based on the assumption that the independent component must have nongaussian distributions [81]. ICA has been used in many video-based HR estimation methods to uncover small PPG signal from the pixel intensity changes of skin/face in the video [27, 28, 31, 37–42, 44, 62, 72, 82]. Poh (and Picard) *et al.* [27, 28] obtained the mean pixel values of each RGB channel (in the facial region) for each frame and used ICA on each RGB signal to estimate the underlying HR signal. Tsouri *et al.* [40] addressed that standard ICA techniques suffer from the sorting problem and used constrained ICA (cICA) to make use of prior knowledge about the underlying sources in VHR. Monkaresi *et al.* [62] extended the method proposed by Poh *et al.* [27] by using k-nearest neighbor (kNN) Machine Learning technique on the ICA outputs. Sahindrakar *et al.* [37] also used ICA for

obtaining PPG signal under limited motion of a subject. Sun *et al.* [38,39] and Zhao *et al.* [31] described a similar approach using ICA but only using a single channel. Esttepp *et al.* [41] captured raw format 120 fps videos from nine imagers under controlled lightings and recovered PPG source component from 9-imager channel space based on ICA method. Yu *et al.* [42] used a combination of ICA and mutual information to compute the dynamic heart rate variation from short video sequence. They defined the mutual information between two sources such that it is zero if both sources are totally independent of each other and unity if both sources are totally dependent on each other. They used this information is used to ensure the reliability of the ICA sources being found [42].

Other approaches include replacing ICA with other linear dimensionality reduction methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA constructs a linear subspace that best explains the variation of observed data from their mean [83]. LDA finds the directions in the underlying vector space that are maximally discriminating between the classes by simultaneously maximizing the between-class scatter and minimizing the within-class scatter [84]. PCA or LDA uncorrelate the source signals without taking care of the non-Gaussianity between the source signals but some papers suggested that they give similar result to ICA-based method while requiring less computation. Lewandowska *et al.* [45] estimated pulse rate from web camera recordings and compared the ICA-based approach with that of PCA-based approach. They suggested PCA requires less computation while giving similar accuracy when compared to ICA-based approach in their experimental settings. Yu *et al.* [46] described that to estimate the instantaneous heart rate that varies dynamically from short video sequences, PCA is less computationally intensive than ICA. Tran *et al.* [47] used LDA to obtain HR signals based on the observation that the fluctuations in RGB traces due to heart pulse have strong correlation between each channel. Their RGB traces are obtained from the skin pixels within the face region. They suggested that LDA can be used for video-based real-time HR estimation because LDA is computationally light.

ICA, PCA and LDA are all based on the assumption that the observed signals are linear mixtures of underlying sources. Wei *et al.* [48] claim that this assumption is wrong because according to Beer-Lambert law, reflected light intensity traveled through facial tissue varies nonlinearly with distance. They present webcam-based HR measurement using a nonlinearity method, Laplacian Eigenmap (LE), addressing neither ICA nor PCA could extract the pure BVP from collected data, as both of them are based on linear hypothesis.

Other approaches have used spatial decomposition and temporal filtering to magnify the video and then find subtle changes in the video [30]. This method can be used to visualize the blood volume changes from the video. In [49] Eulerian video magnification [30] for HR estimation was further investigated and concluded the method is highly affected by aggressive compression and motion. Bal [50] used a wavelet transform based on a denoising method to obtain HR from video recorded by the laptop camera. Xu [64] derived the pulse heart rate signal as a pixel quotient in log space based on a model of light interaction with human skin.

Face/skin areas are used as region of interests (ROI) in many VHR approaches but other various regions including the sub-regions within the face have been explored [25, 35, 38, 44, 45, 51–54, 54, 56, 59, 65–68, 74]. In [69], they placed a finger between a camera of 100 fps and a LED light where the distance between the camera and the LED is from 20 cm to 1.5 meters. Hand palm area was used to obtain noncontact PPG signal in [68]. Lewandowska *et al.* [45] used a rectangular-shaped part of the forehead area. They assumed the forehead areas are visibly “uniform” and took thermographic images for all examined participants to support this assumption. Forehead/brow area was also used in [43, 44, 51, 53, 63, 67, 75]. Several papers used cheek regions [25, 53, 54, 65]. [53] addressed that the “forehead” and “cheeks” are the most desirable ROI due to their relatively robustness to facial expressions and head movement. Rodriguez *et al.* [74] excluded the eye area to eliminate the artifacts produced by blinking. The area below the eyes and above the upper lip of the mouth was used in [56]. Tasli *et al.* [66] used adaptive ROI regions within the face by detecting facial

landmark locations. Other ROIs include sub-dividing the face region into multiple sub-regions [35, 38, 52, 55, 59, 70]. The sub-regions with large intensity variations were rejected and only the regions with small variations were used for HR estimation in [35]. They assumed that large variations are mostly due to illumination change or motion artifacts. Qi *et al.* [70] formed datasets of RGB traces from seven sub-regions around cheek and nose and obtained latent source component of each dataset using joint BSS. [55] focused on improving the PPG signal quality in VHR through a dynamic ROI approach.

Some papers use signals from both the face/skin regions and the background regions [32, 36, 52] to better estimate HR. Li *et al.* [32] assumed that the signals observed from the video were affected by both the PPG signal and environmental illumination. They used both the face mean green color and the background mean green color of each frame to reduce the environmental illumination variances. Tarassenko *et al.* [52] used a background region of interest to minimize the effects of external lighting sources such as fluorescent lights using an Auto-Regressive (AR) model and a pole-cancellation algorithm. Lee *et al.* [73] assumed that the green channel trace from the face region of the video contains both a PPG signal and environmental illuminance change. In their experimental setting, a subject is watching a video in front of a 42-inches monitor in a dark room while the camera is recording the their face. Instead of using the signal from the background region, they estimated the environmental illumination from the face region signal through regression. After estimating the variation of the environmental illuminance, they subtracted it from the green channel trace of the face region [73].

Most of the VHR methods make use of the fact that HR ranges in certain frequency range and this involves the frequency-domain analysis of the observed signal. The most common way to do the frequency-domain analysis is using Discrete Fourier Transform (DFT). Fouladi *et al.* [82] addressed that DFT is not accurate enough to use on small number of samples (2-second length signal for 30 fps case). They



suggested to use the Multiple Signal Classification (MUSIC) method in case of small number of samples.

Instead of using PPG, Balakrishnan *et al.* [33] estimated HR from subtle motion changes captured in the standoff video. The motion changes obtained would reflect BCG signal.

Each of the methods above have challenges. Patients have varying skin tones, a wide range of resting heart-rates, and are often prone to movement. Similarly, environmental lighting, undetermined noise, or camera-based signals can reduce the signal to noise ratio (SNR). Low frequency rate, low video resolution, low video quality and short video length can also cause difficulties in VHR. Greater distance between a subject and the camera can cause lower resolution and quality on the PPG ROI. Shagholi *et al.* [72] experimented on two distances of 0.5 and 3 meters and suggested that increasing the distance to 3 meter led to a decrease in the accuracy of the estimated HR. Small temporal variation corresponding to PPG signal might get corrupted while compressing the video. Kirenko *et al.* [85] described a video encoding/decoding device, wherein during decoding PPG relevant information is preserved. Freitas [77] used raw format in an AVI container file because a video compression algorithm could cause damage to the acquired PPG signal while maintaining the perceived quality of the compressed video. Several other VHR papers also used raw video formats in their experiments along with turning off the automatic white balance (AWB) control or automatic gain control (AGC) of the camera settings. In addition to AWB or AGC, other blocks in camera image signal processing (ISP) pipeline that involves temporal smoothing can reduce the SNR of PPG signals contained in video recordings.

In this thesis, we improve an existing video-based HR estimation method and compare it to an FDA-approved medical device (i.e., a finger pulse oximeter). We modify and extend an ICA-based method and improve its performance by (1) adapting the passband of the bandpass filter (BPF) or the temporal filter, (2) by removing background noise from the signal by matching and removing signals that occur in the

off-target (background) and on-target areas (facial region), (3) face tracking, and (4) skin detection within the face region. Our system is described in Chapter 2.

One of the biggest challenges in video-based HR estimation is dealing with human videos where the subject moves. For the same shooting environment where we can acquire strong HR signals from non-moving subjects, the HR signal gets weaker or even disappears when subjects start to move.

Even in contact PPG, the motion artifacts (MA) has been one of the most challenging problems. Not all the reasons for MA in contact PPG would be the same as VHR case. But there could be a common reason for motion effects in both fields since both contact PPG and VHR estimate the PPG signal from the skin reflectance change using photo sensors. Raghu *et al.* [86] addressed in-band noise results when the spectra of MA and that of the PPG signal overlap significantly. They described that adaptive filters can effectively deal with in-band noise but it needs a reference signal that is strongly correlated with either (1) the artifact but uncorrelated with the signal or (2) the signal but uncorrelated with the artifact. And the reference signal representing MA can be obtained by employing additional hardware [86]. Wijshoff *et al.* [87] addressed that sensor motion relative to the skin can be used as an artifact reference in a correlation canceller to reduce motion artifacts. In their experiment, they obtained sensor motion via self-mixing interferometry. Lee *et al.* [88] investigated the use of red, green, and blue light PPG to discover which of these is the most suitable for measuring HR during normal daily life, where motion is likely to be a significant issue. Based on their experimental results, they concluded that the green light PPG might be more suitable for monitoring of HR in the daily life than either red or blue light PPG. Zhang *et al.* [89] focused on HR monitoring using wrist-type PPG signals when wearers do intensive physical activities. They noted that compared to fingertip and earlobe, wrist can cause much stronger and complicated MA due to large flexibility of wrist and loose interface between pulse oximeter and skin. Hayes [90] addressed the motion model and suggested the multiplicative model is more appropriate for the effect of MA than an additive model based on the experiments.

While most of the MA reduction approaches in contact PPG make use of the reference signal that represent the noise, the approaches in VHR have been more focusing on choosing better ROI in the frame or manipulating RGB traces to enhance the SNR of PPG signal. Kumar *et al.* [35] proposed to track different non-rigid regions of the face independently to compensate for motion-related artifacts. From the fact that blood volume change underneath the skin causes very small changes in the intensity of reflected light signal, they identify the regions with large intensity changes and reject those regions assuming that the large intensity changes have been caused by motions. Feng *et al.* [54] proposed an adaptive color difference method between the green and red channels along with ROI tracking to remove motion artifacts. Motion models used in above two papers [35, 54] will be described in Section 3.1. Wang *et al.* [59] first find temporally corresponding pixels by pixel-based tracking to obtain pixel-to-pixel RGB signal. Then they remove the motion-induced color distortions based on the assumption that the transformation between normalized RGB for consecutive frames should ideally be the translation for the pulse-induced color change while it is not for the motion-induced color change. Huang *et al.* [71] used a similar approach to a MA removal method used in contact PPG. They obtained  $(x, y)$  coordinate of the ROI as the reference signal of the motion and used them as inputs to adaptive filter to reduce the interference related to motion [71]. While many papers address the motion artifacts and give different solutions, there are not enough explanations on how exactly this motion-related signal is generated.

In Chapter 3 we describe the motion-related signal as the change in relative positions between the subject’s skin surface and the light source. We show how the pixel intensity changes are related to motions by showing its relation to the angle between the light ray and the skin surface in case of moving subject. First we use a simple model to understand the pixel intensity changes for moving objects and we extend our observation to the human videos. For the experiment on the human videos, we modeled the pixel intensity in terms of surface normal and the light direction. None of the VHR papers we have seen so far [25–79] used illumination information that can

be acquired from facial points. In our experiment, motion-related signal estimated from the illumination information shows strong relation to the actual intensity variations caused by motion. We extended the experiment to videos of human and showed the motion effects on the intensity change in terms of the skin surface normal and illumination. Our results show how the incident angle change caused by motion is related to the pixel intensity changes. We showed that the illumination change on each surface point is one of the major factors causing motion artifacts. Lastly, we discussed how this understanding on motion effects can be used to reduce the motion artifacts in VHR.

VHR involves recording videos and health information of the human subjects. Research involving human subjects requires an approval of Institutional Review Board (IRB). All the videos that we used in this thesis were collected under the approval of the IRB of Purdue University. Publicly sharing the VHR data in the research community is difficult because this might violate the IRB rules in terms of protecting the privacy of the human research subjects unless the human subject consent on fully disclosing their information public. Some recent VHR papers [32, 36] used publicly available dataset which included video recordings of adult subjects and their ECG signals.

### 1.1.2 Sleep Analysis

Pediatric sleep medicine is a field that focuses on typical and atypical sleep patterns in children. Within this field, physicians, interventionist, and researchers record and label child sleep with particular attention to sleep onset time, total sleep duration, and the presence or absence of night awakenings. One notable recording method is videosomnography (VSG) which includes the labeling of asleep vs. awake from video [2, 3]. This method is most commonly used for infants/toddlers as their compliance rates with other sleep recording methods can be low. Traditional behavioral videosomnography (B-VSG) includes manual labeling of awake and sleep states by

a trained technician/researcher [3]. B-VSG is time consuming and requires extensive training which has limited its widespread use within the pediatric sleep medicine field. Actigraphy is considered an alternative for estimating sleep vs. awake states and it is based on child movement as indexed by an accelerometer sensor commonly attached to a child’s ankle or wrist. Some validity issues with actigraphy compared to human observations are estimating less time sleep and more time awake [91] or showing low specificity in detecting wakefulness within sleep periods [92]. Actigraphy requires a sensor to be constantly attached to the body while VSG and B-VSG do not involve any contact with the body. Within the present study we develop and test an automated VSG method (auto-VSG) to replace B-VSG and to provide physicians, interventionist, and researchers with a sleep recording tool that is more economic and efficient than B-VSG, while maintaining high levels of labeling precision.

The development of auto-VSG is a growing area with preliminary studies utilizing signal processing systems that index movement during sleep in small groups of children with developmental concerns or adults [2, 93–95]. Across these studies, motion within the video is estimated by frame differencing [93, 94] or by obtaining motion vectors [2, 95].

[93] devised a sleep evaluation technique for children by estimating the amount of motions from the difference in two successive frames. They analyzed the relation between the amount of movements obtained from video processing and sleep stage determined by PSG for five children. Their method was later on used in [96] for characterizing the differences in body movements during sleep for eleven typical developing children and six children diagnosed with Attention Deficit/Hyperactivity Disorder (ADHD). [96] suggested that the video-based method may be used as a marker in the diagnosis of ADHD. [94] used variation of image difference processing in [93, 96] by focusing more on the gross body movements (GMs). They used their method to compare the movements during sleep of children with and without ADHD. [2] used home-videosomnography for children with neurodevelopmental conditions where the movement analysis was done using the software called Optical Flow

Algorithm. [95] proposed sleep video motion estimation based on a spatio-temporal prediction method. Their proposed method is to estimate not only the amount of motion but also the direction of movement in order to estimate local motions.

However, each of these studies were completed within a controlled setting and do not account for the wide range of camera positions and lighting variations that are common among in-home VSG recordings.

In this thesis, we present two different sleep video analysis approaches where both uses simple motion information from in-home VSG recordings for children. It is important to note that our goal is to label each frame of a sleep video with the label “sleep” or “awake.” In this work we are not interested in labeling sleep stages, such as REM sleep. We assume that the child is the only source of the movement in the video. Also, we assume that the camera is static. These were the common cases for in-home child sleep videos. While there are other complicated methods for detecting motion in videos, we focus on simple motion information obtained from frame difference method. There are three reasons for choosing simple motion information. First is to make the operation efficient and simple. The amount of sleep videos is massive. For example, one-night video of 8-hour duration recorded at 16 fps includes around 450,000 frames ( $16 \text{ [fps]} \times 60 \text{ [second/minute]} \times 60 \text{ [minute/hour]} \times 8 \text{ [hours]} = 460,800 \text{ [frames]}$ ). When it comes to multiple-night recording on many different children, the processing should be fast and efficient. Second, it is not practical to use complicated methods on low-quality infrared videos. Sleep videos are recorded in either RGB or infrared modes depending on whether the room light is on or not. When in infrared mode, they lack color information. Also, the videos are mostly low-quality where it is good enough for human to identify the movement of the child in the video. The video recordings used for VSG in sleep lab were typically  $320 \times 240$  and  $640 \times 480$  and the images are not sharp. Lastly, the simple motion information captures relative amount of motions within the video very well. While the simple motion information gives useful information for sleep analysis, there are challenges for using it in practical auto-VSG applications. It does not account for ‘in the wild’ factors that are common

in in-home VSG recordings. For example, the wide range of camera positions and lighting variations across different videos make the scale of the motion information different across the videos.

In this thesis, we present two auto-VSG that adjust for these ‘in the wild’ factors. In Chapter 4, we develop and test an auto-VSG method that includes (1) preprocessing the video frames using histogram equalization and resizing, (2) detecting infant movements using simple motion information, (3) estimating the size of the infant by detecting their heads based on deep learning methods, and (4) scaling and limiting the degree of motion based on a reference size so the motion can be normalized to the size of the relative child in the frame. In Chapter 5, we propose automatic sleep/awake states identification methods on RGB/infrared video recordings. It is a binary classification problem for actions in sleep videos. The contributions of this proposed method are: (1) we describe the key factors in sleep video classification (i.e., movements over long period of time) that are not addressed in commonly used action classification problems (Section 5.2) (2) we propose a sleep/awake classification system with a recurrent neural network using simple motion information (Section 5.3) (3) we experimentally show our system successfully learns long-term dependencies in sleep videos and outperform one of the recent method that has been successful in public action dataset (Section 5.4). In Appendix A, we describe web application that deploys our sleep/awake classifications method in Chapter 5 and we call it Sleep Web App. The design philosophy of Sleep Web App is to provide easy accesses to sleep researchers for running the sleep video analysis on their videos. Specifically, we focused on (1) simple user experience, (2) multi-user supporting and (3) providing results for further analysis.

## 1.2 Image-Based Geographical Location Estimation Using Web Cameras

Thousands of sensors are connected to the Internet [97, 98]. The “Internet of Things” will contain many “things” that are image sensors [99–101]. This vast net-

work of distributed cameras (i.e. web cams) will continue to exponentially grow. We are interested in how these image sensors can be used to sense their environment.

In our previous work, we investigated simple methods of web cam image classification based on the support vector machine (SVM). We focused on classifying an image as indoor or outdoor and people or no people using a set of simple visual features.

We are also investigating how one would process imagery from thousands of ip-connected cameras. We have at Purdue University been developing the CAM<sup>2</sup> system (Continuous Analysis of Many CAMeras) [102–105]. CAM<sup>2</sup> is a cloud-based general-purpose computing platform for domain experts to extract insightful information by analyzing large amounts of visual data from distributed sources. CAM<sup>2</sup> uses cloud computing to manage the large amounts of data for better scalability. CAM<sup>2</sup> currently has detected and has access to more than 70,000 cameras deployed worldwide. These include cameras from departments of transportation, national parks, research institutions, universities, and individuals.

In particular in this thesis we investigate simple methods for how one can determine metrics of a location (e.g. sunrise/sunset, length of day) and the location of the web camera by observing the camera output.

The location of a point on the Earth is described by its latitude and longitude (and perhaps by its altitude above sea level). Latitude is measured in degrees north or south of the Equator, 90° north latitude is the North Pole and −90° south latitude is the South Pole. Longitude is measured in degrees east and west of Greenwich, England. 180° east longitude and −180° west longitude meet and form the International Date Line in the Pacific Ocean [106–108]. The definition of sunrise and sunset is when the geometric zenith distance of the center of the Sun is 90°50′ [109]. That is, the center of the Sun is geometrically 50 arcminutes below a horizontal plane. There are various definitions for sunrise/set and daylength [110].

Several approaches have been reported with respect to finding a location from images using large database. Hays *et al.* [111] described a method to estimate geographic information from a single image using a purely data-driven scene matching



approach. They used a dataset of over 6 million GPS-tagged images from the Internet. The features they used for comparing the images are color image itself, color histogram in CIE  $L^*a^*b^*$  color space, texton histogram, line features, Gist descriptor together with color, and geometric context [111].

Sunkavalli *et al.* [112] model the temporal color changes in outdoor scenes from time-lapse video to provide partial information of scene and camera geometry regarding the orientation of scene surfaces relative to the moving sun. With assumptions that reflectance at scene points is Lambertian, and that the irradiance incident at any scene point is entirely due to light from the sky and the sun, they came up with a model for temporal intensity change in terms of the angular velocity of the sun and the projection of the surface normal at a scene point onto the plane spanned by the sun directions (the solar plane) along with other factors. They estimated camera geo-location, latitude and longitude, from the image sequence of one building scene captured over the course of one day with approximately 250 seconds between frames. This method requires three scene points lying on three mutually orthogonal planes (two sides of a building and the ground plane for example) in the image. Lalonde *et al.* [113] used high-quality image sequence to estimate camera parameters. In order to do this, they analyze the sun position and the sky appearance within the visible portion of the sky region in the image. Then, from an equation expressing the sun zenith and azimuth angles as a function of time, date, latitude and longitude, they estimated the latitude and longitude of the camera.

Junejo *et al.* [114] geo-located the camera from shadow trajectories estimated from image sequence. Latitude was estimated based on the fact that the path of the sun, as seen from the earth, is unique for each latitude [114]. They estimated the longitude from the local time stamp of the image and shadow points. In their experiment, they selected the shadow points of a lamp post and a traffic light on the images. Wu *et al.* [115] also described camera geo-location estimation based on two shadow trajectories. They employed a semi-automatic approach to detect the shadow point for an input video.

Sandnes [116] estimated approximate geographical locations of webcams from sequence of images taken at regular intervals. First, the sunrise and sunset were estimated by classifying images taken from a webcam and the location was then estimated [116]. For determining the sunrise and sunset, the intensity of the entire image was used to classify day or night and then determine the midday (or local noon) time to identify the longitude and latitude [116]. In this thesis, we modify and extend Sandnes’s approach.

We used the the sky regions in the image to better classify the Day/Night images. Several papers described methods for detecting sky regions [117–119]. In [117] the sky region is identified by using image data taken under various weather conditions, predicting the solar exposure using a standard sun path model, and then tracing the rays from the sun through the images. In [118] vehicle detection and tracking is used to detect road conditions in both day and night images by using images and sonar sensors. A method to retrieve the weather information from a database of still images was presented in [119]. The sky region of image was detected by using the difference of pixel values from successive image frames, morphological operations were then used to obtain a sky region mask. The weather condition was recognized by using features such as color, shape, texture, and dynamics.

In this thesis we describe a method for estimating the location of an IP-connected camera (a web cam) by analyzing a sequence of images obtained from the camera. First, we classify each image as Day/Night using the mean luminance of the sky region. From the Day/Night images, we estimate the sunrise/set, the length of the day, and local noon. Finally, the geographical location (latitude and longitude) of the camera is estimated. The system is described in Chapter 6.

### 1.3 Contributions of This Thesis

The main contributions of this thesis are listed as follows:

- We improved VHR for assessing resting HR in a controlled setting where the subject has no motion. We modified and extend an ICA-based method and improve its performance by (1) adapting the passband of the bandpass filter (BPF) or the temporal filter, (2) by removing background noise from the signal by matching and removing signals that occur in the off-target (background) and on-target areas (facial region), and (3) detect skin pixels within the facial region to exclude pixels that does not contain HR signal.
- We investigated and described the motion effects in VHR in terms of the angle change of the subject’s skin surface in relation to the light source. We showed that the illumination change on each surface point is one of the major factors causing motion artifacts by estimating the incident angle in each frame. Based on this understanding, we discussed the future work on how we can compensate for the motion artifacts.
- We proposed auto-VSG method where we used child head size to normalize the motion index and to provide an individual motion maximum for each child. We compared the proposed auto-VSG method to (1) traditional B-VSG sleep-awake labels and (2) actigraphy sleep vs. wake estimates across four sleep parameters: sleep onset time, sleep offset time, awake duration, and sleep duration. In sum, analyses revealed that estimates generated from the proposed auto-VSG method and B-VSG are comparable.
- In the next proposed auto-VSG method, we described an automated VSG sleep detection system which uses deep learning approaches to label frames in a sleep video as “sleep” or “awake” in young children. We examined 3D Convolutional Networks (C3D) and Long Short-term Memory (LSTM) relative to motion information from selected Groups of Pictures of a sleep video and tested temporal window sizes for back propagation. We compared our proposed VSG methods to traditional B-VSG sleep-awake labels. C3D had an accuracy of approximately 90% and the proposed LSTM method improved the accuracy to more than 95%.

The analyses revealed that estimates generated from the proposed LSTM-based method with long-term temporal dependency are suitable for automated sleep or awake labeling.

- We created web application (Sleep Web App) that makes our sleep analysis methods accessible to run from web browsers regardless of users' working environments. The design philosophy of Sleep Web App is to serve easy accesses to sleep researchers for running the sleep video analysis on their videos. Specifically, we focused on (1) simple user experience, (2) multi-user supporting and (3) providing results for further analysis. For providing the results, we included two csv format files for per-minute sleep analysis and sleep summary results.
- We also described a method for estimating the location of an IP-connected camera (a web cam) by analyzing a sequence of images obtained from the camera. First, we classified each image as Day/Night using the mean luminance of the sky region. From the Day/Night images, we estimated the sunrise/set, the length of the day, and local noon. Finally, the geographical location (latitude and longitude) of the camera is estimated. The experiment results show that our approach achieves reasonable performance.

#### 1.4 Publications Resulting From This Thesis

1. **J. Choe**, A. J. Schwichtenberg, E. J. Delp, "Classification of Sleep Videos Using Deep Learning," *Proceedings of the IEEE Multimedia Information Processing and Retrieval*, pp. 115–120, March 2019, San Jose, CA.
2. A. J. Schwichtenberg, **J. Choe**, A. Kellerman, E. Abel and E. J. Delp, "Pediatric Videosomnography: Can signal/video processing distinguish sleep and wake states?," *Frontiers in Pediatrics*, vol. 6, num. 158, pp. 1-11, May 2018.
3. **J. Choe**, D. Mas Montserrat, A. J. Schwichtenberg and E. J. Delp, "Sleep Analysis Using Motion and Head Detection," *Proceedings of the IEEE Southwest*

*Symposium on Image Analysis and Interpretation*, pp. 29–32, April 2018, Las Vegas, NV.

4. D. Chung, **J. Choe**, M. OHaire, A. J. Schwichtenberg and E. J. Delp, “Improving Video-Based Heart Rate Estimation,” *Proceedings of the Electronic Imaging, Computational Imaging XIV*, pp. 1–6(6), February, 2016, San Francisco, CA.
5. **J. Choe**, D. Chung, A. J. Schwichtenberg, and E. J. Delp, “Improving video-based resting heart rate estimation: A comparison of two methods,” *Proceedings of the IEEE 58th International Midwest Symposium on Circuits and Systems*, pp. 1–4, August 2015, Fort Collins, CO.
6. T. Pramoun, **J. Choe**, H. Li, Q. Chen, T. Amornraksa, Y. Lu, and E. J. Delp, “Webcam classification using simple features,” *Proceedings of the SPIE/IS&T International Symposium on Electronic Imaging*, pp. 94010G:1–12, March 2015, San Francisco, CA.
7. **J. Choe**, T. Pramoun, T. Amornraksa, Y. Lu, and E. J. Delp, “Image-based geographical location estimation using web cameras,” *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 73–76, April 2014, San Diego, CA.

## 2. PROPOSED APPROACH FOR RESTING HEART RATE ESTIMATION

### 2.1 Overview of Proposed System

Figure 2.1 shows our proposed method and is similar to the ICA-based method described by Picard in [28, 29]. The gray blocks denote modifications/additions to Picard’s approach [28, 29] described below. We will present a brief overview of Picard’s method, more detail is available in [28, 29]. The ‘Picard’ ICA-based method begins by detecting the face region. For each face region, the mean RGB pixel value is obtained across each frame to form three 1D time series signals we call the RGB traces. Trends in the RGB traces due signal drift and other factors are then removed by using a high-pass like detrending technique [120]. The cutoff frequency of this filter is controlled by a parameter we denote as  $\lambda$ , where  $\lambda = 300$  in our experiments. This corresponds to a high pass cutoff frequency of  $0.011 \cdot f_s$  Hz where  $f_s$  is the sampling rate where  $f_s = 30$  Hz (the videos are acquired at 30 frames/s). The detrended traces are normalized with z-score normalization to produce zero-mean and unit variance signals. Independent Component Analysis (ICA) is used on these three signals to recover the target source signal [28, 29, 81].

The appropriate source signal is selected from the ICA output by computing the normalized Power Spectral Density (PSD),  $P[k]$  with  $k$  the frequency index and choosing the component that has the highest peak of PSD within the frequency range  $f_l = 0.7$  and  $f_h = 3$  Hz. Where  $f_l$  and  $f_h$  are the fixed cutoff frequencies for the range of all possible HR [28, 29]. After a five-point moving average filter ( $M = 5$ ), the signal is bandpass filtered with a 128-point Hamming window (filter order  $N_f = 127$ ) and with cutoff frequency of  $f_l$ - $f_h$  Hz. This is the same frequency range used in the PSD/Highest Peak block. Next the signal is interpolated to the new sampling rate of

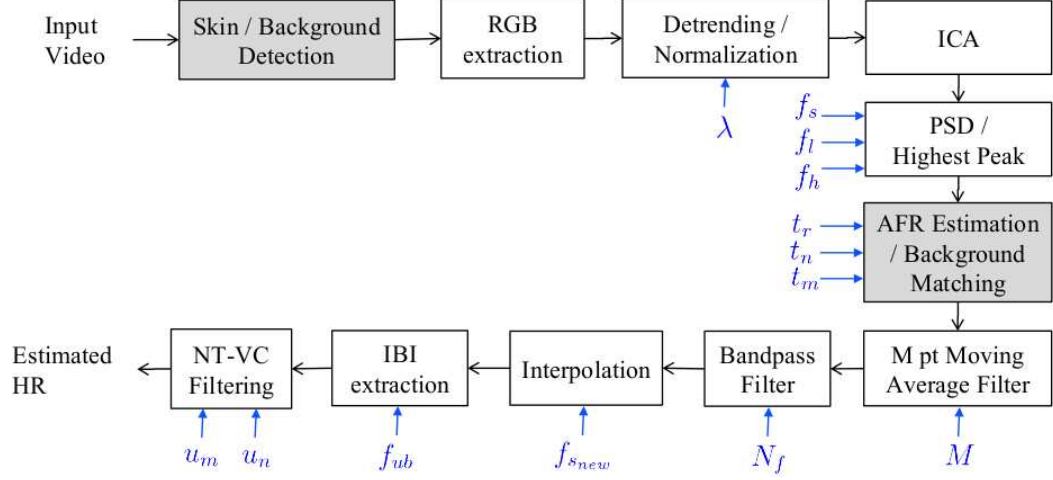


Fig. 2.1. The block diagram of the proposed system (after [28, 29]).

$f_{s_{new}} = 256$  Hz. To find the HR in units of bpm, first the Inter-Beat Interval (IBI) is obtained from the interpolated signal. IBI is the time intervals between the peaks in units of seconds. The peaks are all the values that are the largest inside the sliding windows [28, 29]. The window size,  $T_w$  [sec], is a parameter for the IBI and should be smaller than the smallest peak interval that  $T_w \leq (1/f_{ub})$  where  $f_{ub}$  is the upper bound frequency for IBI. We use  $f_{ub} = f_h$  in our work. From the maximum value of  $T_w$  and the sampling frequency  $f_{s_{new}}$ , we can obtain the number of points to examine before and after the current point

$$p = \left\lceil \frac{T_w \cdot f_{s_{new}}}{2} \right\rceil \quad (2.1)$$

to determine peaks. By using  $p$  in Eq.(2.1) we can obtain IBI in units of seconds [28, 29]. The reciprocal of each IBI value is then the HR estimates in unit of Hz. Finally, the signal is filtered through the noncausal of variable threshold (NT-VC) filter [28, 29, 121] with fixed parameters  $u_n = 0.4$ , and  $u_m = 1.0$  Hz. Unstable HR estimates are removed in this final process.

The performance of this method heavily depends on parameter settings and recording environments. Among the parameters, the passband frequency range of the band-pass filter plays a crucial role in estimating HR. In our proposed method we find the

passband frequency range and adapt by observing periodic signals that are generated from the recording environment.

To estimate the HR from video we isolate the subtle changes of blood flow in the face region. There could be many signal sources that contribute to color intensity variations. Since what we want to obtain is the “HR signal,” adapting the passband frequency range is a key factor in HR estimation. The previous work uses a fixed passband frequency range,  $f_l$ - $f_h$  Hz, for the band pass filter (BPF). In our work we estimate the HR signal by adapting the passband frequency range for each participant. We call this the adaptive frequency range (AFR) and denote it as  $f_{a_l}$ - $f_{a_h}$ . The basic idea is to select the passband frequency range of the face region by excluding the background signals. Our model follows several assumptions. The heart rate of an adult ranges from 42 bpm to 180 bpm (0.7 to 3.0 Hz) and does not change dramatically over time. We assume that IBI will change no more than 2.5 sec (24 bpm). While there are other periodic signals present due to the scene illumination or camera vibration, we assume one of the strongest periodic signals that appears on the face is microblushing (or the HR signal).

Our approach can be used for the BPF in ICA-based HR estimation [28, 29] and for the temporal filter in video magnification [30, 122]. Our approach begins by detecting both face and background regions. Two sets of RGB traces (6 1D signals) from both regions go through the Detrending/Normalization, and then each set (3 1D signals per a set) goes through ICA and PSD/Highest Peak process. In theory, ICA finds the underlying sources that are statistically independent, or as independent as possible from the observed signals [81]. If the output of ICA components are completely independent, we can take one of them to be the HR signal. In practice, we found that several strong periodic signals tend to appear together in the highest peak component. To find only the periodic signal of our interest, we estimate the AFR by using a background matching method and filter out the background matching frequency clusters we describe below.



## 2.2 Frequency Clusters

After the PSD/Highest Peak block in Figure 2.1, we have PSDs both from the face region and the background region. If several periodic signals appear in the face region PSD, we assume one of the periodic signals reflects blood flow changes or an index of HR. To separate the HR signal from the other periodic signals we assess clusters in the frequency domain. A frequency cluster is a continuous range of neighboring frequencies that are generated by thresholding the PSD  $P[k]$  as shown in Figure 2.2. We denote a cluster  $c_i$  by the frequency range  $[f_{i_l}, f_{i_h}]$  where  $i$  is an index of cluster (Figure 2.2). The following three steps show how the clusters are formed.

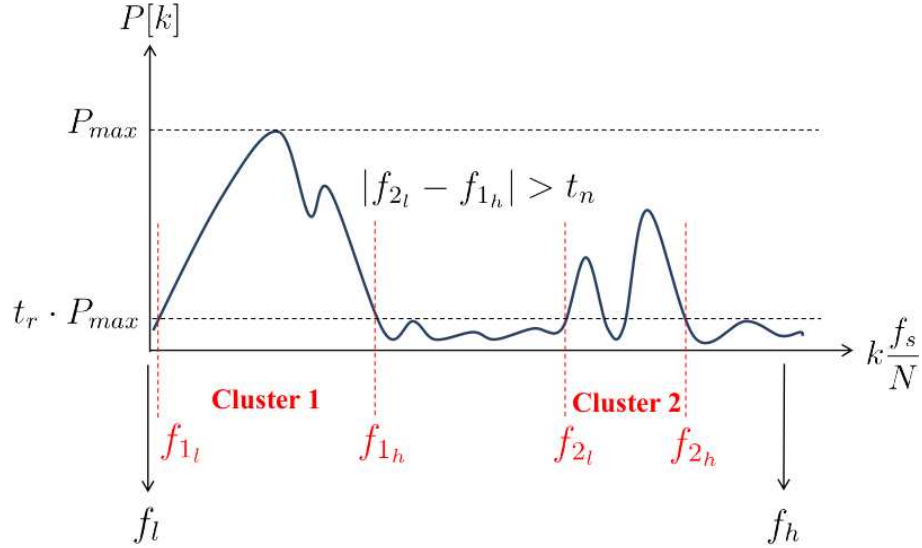


Fig. 2.2. An example of frequency clusters.  $P_{max}$  is the maximum value of the PSD within  $[f_l, f_h]$ .  $t_r$  is a parameter used to determine the weak signals as described in Section 2.2.  $t_n$  is a parameter used to determine the neighboring clusters as described in Section 2.2. If two clusters formed by thresholding  $P[k]$  are with  $t_n$  Hz of one another we considered them ‘neighbors’ and merge them into one cluster. Cluster 1 and Cluster 2 in this Figure are not ‘neighbors’ because  $|f_{2_h} - f_{1_l}| > t_n$ .  $N$  is the number of points in the positive frequency domain, and  $k$  is the index in the frequency domain,  $f_s$  is the sampling rate.

1. Suppress weak signals—weak signals are ignored when forming the clusters. If  $P[k] < t_r \cdot P_{max}$  then weak signal,  $t_r$  is used to determine the weak signal threshold. We empirically choose  $t_r = 0.15$  (15%).
2. Form clusters—repeatedly merge the clusters if two clusters are neighbors.  $t_n$  [Hz] is used to determine the neighboring clusters where  $t_n = 0.1$  Hz (6 bpm) is empirically chosen (see Figure 2.2).
3. Obtain the energy of each cluster (the sum of  $P[k]$  within the cluster).

### 2.3 AFR Estimation by Background Removal

Background removal was achieved by observing both PSDs from face and background regions, we can eliminate frequency clusters in the face region that are similar to the frequency clusters in the background. We measure the shape similarity between two clusters by computing the Sum of Absolute Differences (SAD) between the two normalized PSDs.

$$d = \sum_{k=0}^{n-1} |P_1[k] - P_2[k]| \quad (2.2)$$

where  $P_1$  is the PSD of cluster 1 where the energy is normalized to 1 and  $P_2$  is the PSD of cluster 2 with the energy normalized to 1. The clusters are normalized; therefore,  $0 \leq P_i[k] \leq 1$  and  $0 \leq d \leq 2$ . If the SAD between two normalized clusters is small,  $d < t_m$ , for a parameter  $t_m$ , we deemed that two clusters are similar. The method for AFR estimation is shown below. The estimated AFR is used for the lower and upper bound of the BPF.

1. Go through the first 5 blocks shown in Figure 2.1 to get PSDs for a face and a background region (Section 2.1).
2. Form frequency clusters on each component (Section 2.2).
3. Sort the face frequency clusters based on the energy.

4. Starting from the highest energy cluster of the face signal, select one cluster  $c_{i^*}$  that does not match with any background cluster: we choose  $c_{i^*}$  only if  $d > t_m$  holds between  $c_{i^*}$  and all the background clusters.
5. Obtain AFR from the cluster  $c_{i^*}$  selected in the previous step:  $f_{a_l} = \max(f_{i_l^*}, f_l)$  and  $f_{a_h} = \min(f_{i_h^*}, f_h)$ .

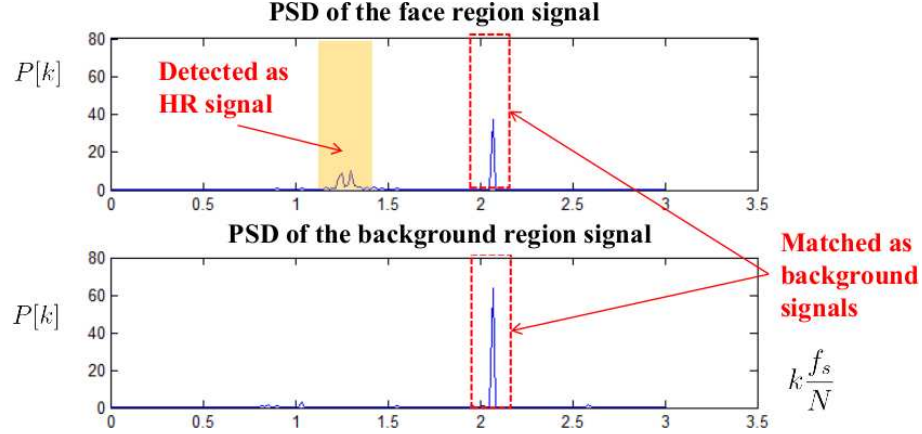


Fig. 2.3. An example of matching clusters from the face signal (top) and the background signal (bottom).  $P[k]$  is the PSD of the signals and  $k$  is the index in the frequency domain.

There can be a corner case where the only frequency cluster matches to the background frequency cluster. This happens when the HR signal from the facial region is not strong enough to form a frequency cluster. In this case, we choose AFR by excluding the background signal to at least get rid of the noise from the background. If there are two different frequency ranges outside of the frequency range for the background signal, we choose the one with the broader range.

## 2.4 Face Tracking and Skin Detection

For tracking, we derived a reference color model from the initial bounding box obtained from the face detection [123] in the first frame. For the color model, each RGB color space is quantized from the original 256 bins to 16 bins and is mapped

into 1D  $16^3$ -bin histogram. The sum of this histogram is then normalized to one. Particle filter tracking is used to find the corresponding face region in each frame [124]. Denoting the hidden state and the data at time  $t$  by  $x_t$  and  $y_t$  respectively, the probabilistic model we use for tracking is

$$p(x_{t+1}|y_{0:t+1}) \propto p(y_{t+1}|x_{t+1}) \int_{x_t} p(x_{t+1}|x_t)p(x_t|y_{0:t})dx_t \quad (2.3)$$

where  $p(y_{t+1}|x_{t+1})$  is the likelihood model of data, and  $p(x_{t+1}|x_t)$  is transition model of second-order auto-regressive dynamics [124]. We define the state at time  $t$  as location in 2D image represented as pixel coordinates. For obtaining the likelihood  $p(y_t|x_t)$ , we use the distance metric  $d(y) = \sqrt{1 - \rho(y)}$  where  $\rho(y)$  is the sample estimate of the Bhattacharyya coefficient between the reference color model and the candidate color model of each particle at position  $y$  [125].

For each pixel within the tracking region, we use skin detection method to exclude non-skin pixels that represent hair, eye or part of the background that do not reflect HR signal. We use the skin classifiers based on Bayes theorem [126] with some variations. [126] made generic skin color model from skin dataset using simple histogram learning technique. The particular RGB value is classified as skin if

$$\frac{P(rgb|skin)}{P(rgb|nonskin)} \geq \Theta, \quad (2.4)$$

where  $0 \leq \Theta \leq 1$  is a threshold and can be written as

$$\Theta = C \frac{P(nonskin)}{P(skin)} \quad (2.5)$$

where  $C$  is the application-dependent parameter [126]. The appropriate value for this parameter differ for various skin tones or lighting conditions. In our system, the user selects the parameter  $C$  from the first frame by moving the track bar and then the selected value is used for the rest of the frames in the video.

[126] suggested to use the linear quantization on each histogram since too many color bins lead to over-fitting while too few bins results in poor accuracy. In their study, they showed that the histogram of size 32 bins/channel gave the best performance when compared to the size 256 or 16. This linear quantization on histograms

for making the skin and non-skin probability models may produce many empty bins in the output histograms. The skin classifications on empty color bins have meaningless results that if we can reduce the number of empty color bins in the quantization step we can obtain better classification performance. In our skin detection method, we create a color-mapping look-up table by adaptively quantizing the histogram using histogram equalization. The goal of histogram equalization is to obtain a uniform histogram [127]. By using the histogram equalization on RGB histograms for skin pixels of training dataset, we map the original RGB color levels to color levels that best represents the skin colors in the training dataset.

Figure 2.4 shows the quatization results for color histogram trained on skin pixles of the publicly available skin dataset [128]. Table 2.1 shows the number of non-empty

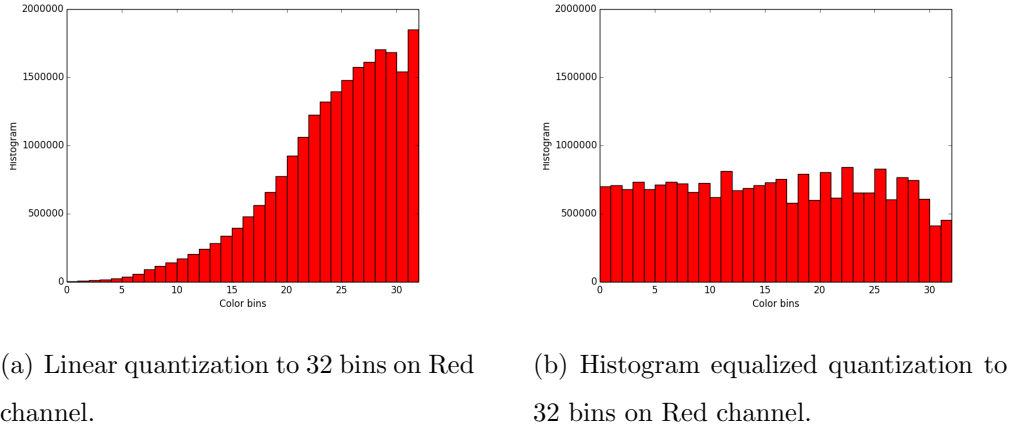


Fig. 2.4. Comparison of two different quantizations on skin pixels.

bins for two different quatizations. The number of empty bin reduced after applying histogram equalization in the quatization step.

Since pixel-based classifier can introduce some falsely classified pixels, we need to refine the result by applying Morphological filtering. Figure 2.5 shows the block diagram of the proposed skin detection system.

Table 2.1.  
Number of non-empty bins for  $32^3$  color bins in UCSB dataset.

|                                     | Skin [bins]       | Non-skin [bins]   |
|-------------------------------------|-------------------|-------------------|
| Linear quantization                 | 8638<br>(26.36%)  | 24484<br>(74.72%) |
| Histogram-equalized<br>quantization | 19428<br>(59.29%) | 30823<br>(94.06%) |

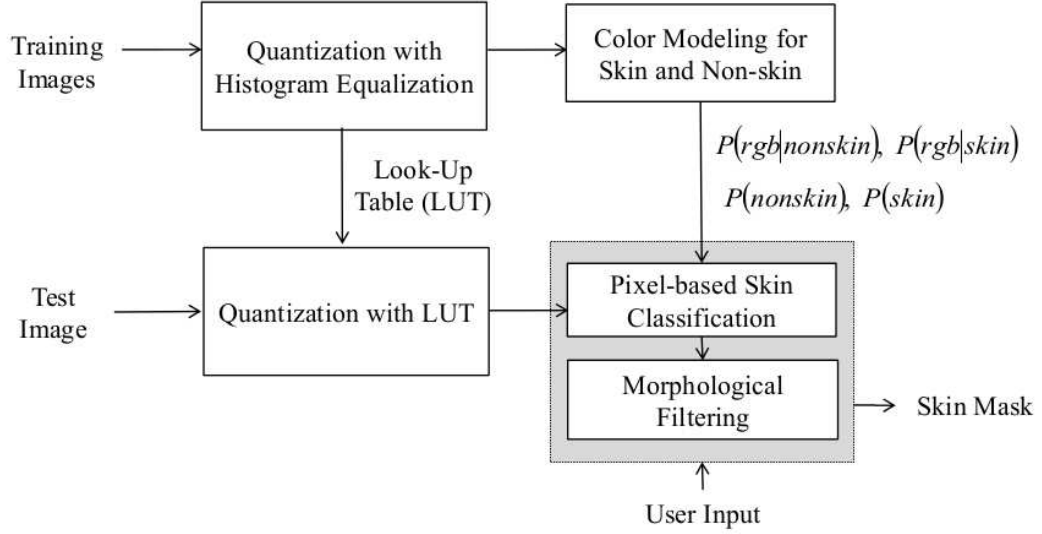


Fig. 2.5. The block diagram of the proposed skin detection system.

## 2.5 Experimental Results

In our experiments we acquired videos of various participants with spatial resolution of  $1920 \times 1080$  and 29.97 fps. There were 22 participants (12 females and 10 males) in Dataset 1 and 18 participants (9 females and 9 males) in Dataset 2 with ages ranging from 20 to 50 years of age. The total number of different people in the entire Dataset is 26 with 14 overlapping participants between Dataset 1 and Dataset 2. The participant numbering for Dataset 1 and 2 are not consistent to each

other. Within the Dataset 2, the participant numbering is consistent for each different task. The data collection methods were approved by the Institutional Review Board of Purdue University. The distance between the participant and the camera was approximately 1.8 m. In Dataset 1, the zoom was manually adjusted to focus on the upper torso and face and Dataset 2 was more zoomed out to show entire upper body as shown in Figure 2.7. Dataset 1 only included no-motion videos that the participants were seated with their arms on the table and were asked to sit still and look toward the camera. Dataset 2 included both no-motion and non-random motion videos. For the non-random motion tasks, the participants were asked to move their head from left to right repeatedly while facing toward the camera. The room had windows with semi-transparent blinds and lighting on the ceiling as shown in Figure 2.6. The ground truth HR was measured using a Nonin *GO<sub>2</sub>* Achieve Finger Pulse Oximeter for Dataset 1 and CE & FDA approved Handheld Pulse Oximeter (model name CMS60D) for Dataset 2. For both cases, the probe was attached to a finger tip of the participant. The output of the pulse oximeter was simultaneously recorded with the face and the two video streams were merged as shown in Figure 2.7. The pulse oximeter HR estimates were manually recorded from the combined video once per second. During the data collection, each participant was asked to select one of the colors in the PANTONE SkinTone Guide [129] that best matches with the skin tone. The PANTONE SkinTone Guide Lightness Scale ranges from 1 to 15 where the scale 1 is the brightest. Within this study, participant skin tones ranged from 1 to 10.

The videos were analyzed offline and from the first frame of each video, the facial region was detected using the OpenCV library [123] with the parameter of minimum face size set to  $120 \times 120$ . With the initial face box in the first frame, tracking box was obtained for rest of the frames in the video. For the Picard’s method, we used the center 60% width and full height of the face/tracking box. For our proposed method, we detected skin pixels within the entire box. The average number of pixels detected as skin within the face region for each participant ranged from



Fig. 2.6. Data Collection Environment.



(a) Dataset 1.

(b) Dataset 2.

Fig. 2.7. Examples of Video Settings.

29,436 to 87,624 for Dataset 1 and ranged from 5,867 to 21,977 for Dataset 2. Our background region requirements were as follow: (1) the area did not contain skin or micro-blushing, (2) the area was not out-of-focus and (3) the size was selected as the 20% width and 50% height of the detected face. We used the video length of 59 seconds for Dataset 1 and 1 minute for Dataset 2. The Joint approximate diagonalization of eigenmatrices (JADE) method [130] was used for the ICA implementation. For the background removal, we used the parameters:  $t_r = 0.15$ ,  $t_n = 0.1$  [Hz],  $t_m = 0.4$ . Selecting appropriate  $t_r$  and  $t_m$  values is crucial. We would like to note that



we used different  $t_r$  value for AFR in our previous work [131] where the difference between our current work were smaller amount of dataset and no skin detection being used.  $t_m$  is a threshold for determining the matching between the foreground and background signals. If the value of  $t_m$  is too low, the background removal process will fail. For this study, SAD ranged from 0 to 2,  $t_m = 0.4$ ; therefore, we determine two frequency clusters were the same if they only differed by 20%. In our recent work [132], we obtained cutoff frequencies by Color Frequency Search (CFS). The advantage of using CFS is that there are less parameters compared to that of AFR and gives tighter cutoff frequencies for the dominant HR value. Disadvantage of CFS is that it has a possibility to miss some of the HR frequency range if HR variance is not low enough to form a dominant peak in the frequency domain. Feng *et al.* [54] used Adaptive Bandpass Filter (ABF) by always setting the cutoff frequency ranges of  $\pm 0.15$  Hz ( $\pm 9$  bpm) around the most dominant peak. Their work only requires one parameter ( $\pm 0.15$  Hz) in terms of setting the adaptive cutoff frequencies but it gives fixed frequency range regardless of the variance of the HR and cannot take care of the background noise.

The initial frequency range to acquire AFR was set to  $f_l = 0.7$  and  $f_h = 3.0$  [Hz]. Resting HR for 95% of healthy adults falls within 48 to 100 bpm (equivalent to  $[0.8, 1.67]$  Hz) [5]. We did not have health information on our participants that we expanded our initial frequency range to  $[0.7, 3.0]$  Hz. Picard's methods [27, 28] used  $[0.75, 4.0]$  Hz or  $[0.7, 4.0]$  Hz.

Figure 2.8 show the Adaptive Frequency Ranges (AFR) for the 22 test cases in Dataset 1. From the figure, we can see that for all participants, the obtained AFR range around their ground truth HR giving much narrower HR range compared to the Fixed HR range. Only Test 13 shows some deviation from GTHR in AFR.

The results using Picard's approach and using our method are shown in Table 2.2. To evaluate the performance, we used the "percentage of acceptance" in NC-VT filter. This is shown in "AccRate" column in the table. Higher acceptance ratios were indicative of more reliable estimates for the obtained estimation. Our second

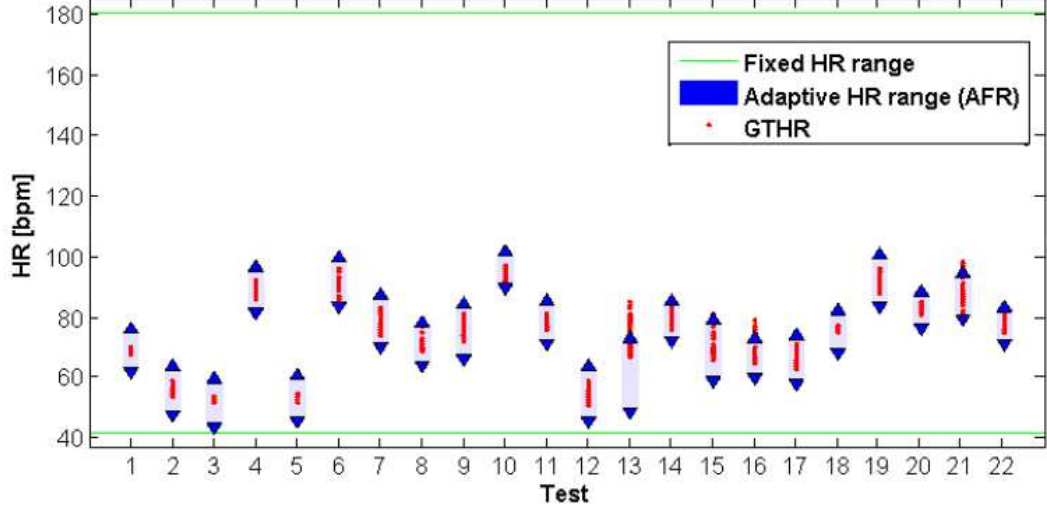


Fig. 2.8. AFR obtained by the Proposed method for Dataset 1.

metric for evaluating the performance was average HR error shown in the “Error” column of the table. HR error is defined as

$$\mu_E = \frac{1}{N'} \sum_{n'} |h[n'] - g[n']| \quad (2.6)$$

where  $h[n']$  is the estimated HR in units of bpm,  $g[n']$  is manually recorded HR at every “second” from the pulse oximeter,  $n'$  is the time domain index for accepted HR estimate and  $N'$  is the number of accepted HR estimates. The “Average GTHR” column is the average value of  $g[n']$ . Our approach has an average  $\mu_E$  3.47 bpm which is notably lower than the 18.76 bpm of the Picard approach. For dataset 1, our HR estimation tends to give less errors for participants with lighter skin tones. For 8 participants with skin tone level 1, the average of  $\mu_E$  is 2.86 bpm and for the rest of the participants with skin tone level ranging from 3 to 8, the average of  $\mu_E$  is 3.83 bpm. Figure 2.9 illustrates the advantages of the AFR for test participant 18.

Figure 2.10 shows the AFR for 18 different test cases in Dataset 2, No-motion videos. The obtained AFR range around their ground truth in most test cases. Test 11 and 13 were the corner cases described in section 2.3 where there is no frequency cluster formed around the ground truth due to weak HR signals. For Motion videos

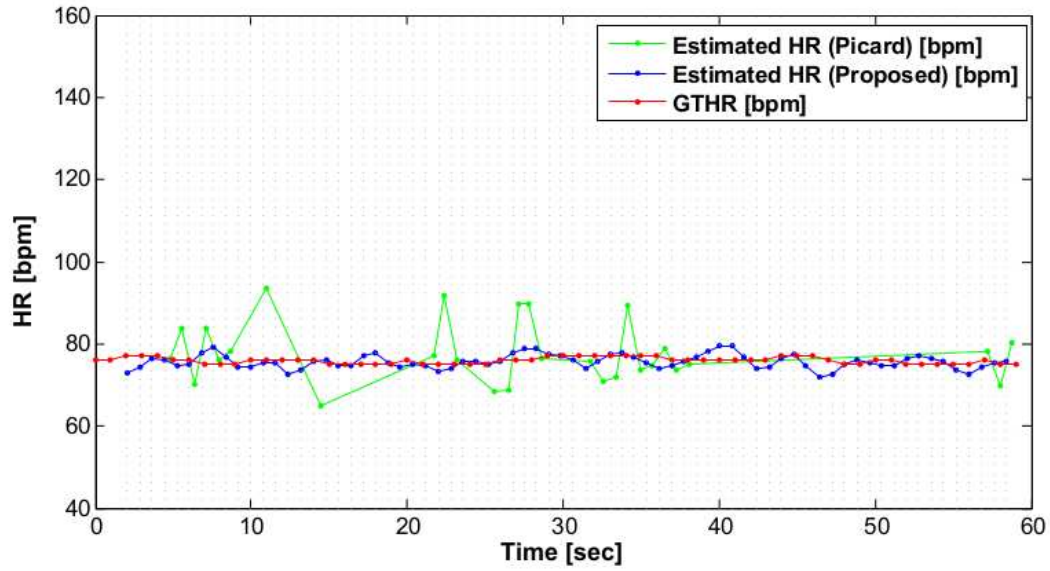


Fig. 2.9. Estimated HRs and Ground Truth HR for Test 18 in Dataset 1.

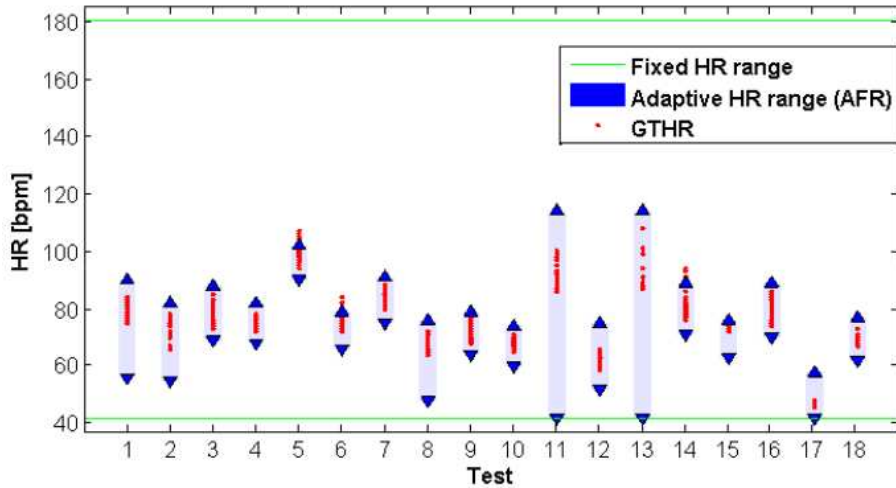


Fig. 2.10. AFR obtained by the Proposed method for Dataset 2, No-motion videos.

in Dataset 2 in Figure 2.11, only half of the test cases have AFRs around their ground truth. Table 2.3 and 2.4 show the results for Dataset 2. For No-motion videos shown in table 2.3, our approach has an average  $\mu_E$  4.87 bpm which is much lower than the 18.04 bpm of the Picard approach. For test 11 and 13, the AccRate is far below 80% and the  $\mu_E$  is high. For these cases, the signal corresponding to the HR was not

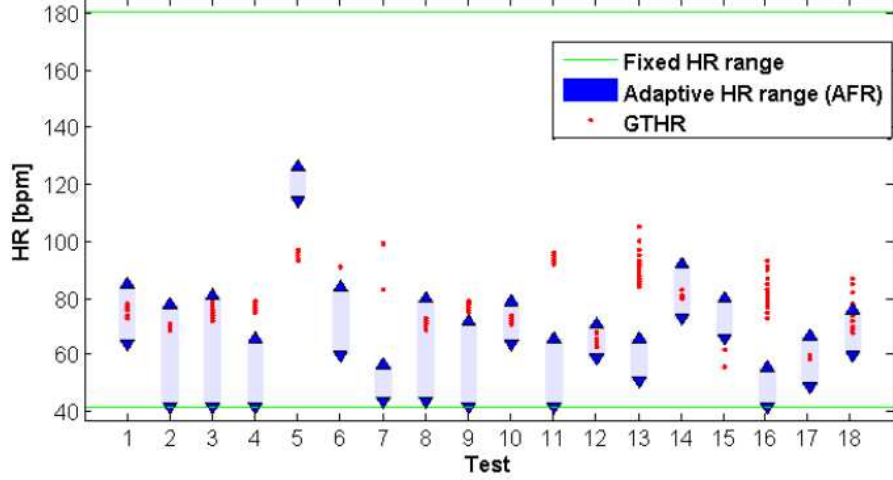


Fig. 2.11. AFR obtained by the Proposed method for Dataset 2, Motion videos.

strong enough compared to other unknown noises. The skin tones did not seem to have strong relationship with the HR estimation error rates in Dataset 2.

For non-random motion videos shown in table 2.4, neither Picard’s approach nor our proposed method gave good HR estimations. For the proposed method, only 4 out of 18 tests showed reasonable HR estimates—AccRate higher than 85% and  $\mu_E < 5$ . Lots of strong signals are generated for motion videos that our proposed method failed to correctly estimate HR for those motion videos.

In sum, we improved video-based methods for assessing resting HR in a controlled setting where there is no motion in the video. We demonstrated that our method can estimate HR with 3.55 bpm for Dataset 1 and 4.87 bpm for Dataset 2 (smaller facial region than Dataset 1) of errors on average across participants with varying skin tones. We will discuss about these motion-generated signals in Chapter 3.

Table 2.2.  
A Comparison of Two Methods for Dataset 1

| Test        | Skin<br>Tones | Picard's approach [28, 29] |               | Proposed Method |               |
|-------------|---------------|----------------------------|---------------|-----------------|---------------|
|             |               | AccRate [%]                | $\mu_E$ [bpm] | AccRate [%]     | $\mu_E$ [bpm] |
| 1           | 6             | 30                         | 14.03         | 97              | 1.25          |
| 2           | 7             | 37                         | 6.92          | 100             | 2.38          |
| 3           | 1             | 13                         | 46.97         | 96              | 3.28          |
| 4           | 1             | 52                         | 7.29          | 98              | 1.97          |
| 5           | 1             | 3                          | 7.18          | 100             | 1.92          |
| 6           | 5             | 26                         | 12.81         | 98              | 4.11          |
| 7           | 1             | 8                          | 27.35         | 100             | 2.77          |
| 8           | 1             | 26                         | 9.87          | 97              | 2.57          |
| 9           | 3             | 12                         | 7.13          | 97              | 2.23          |
| 10          | 5             | 48                         | 25.05         | 98              | 2.63          |
| 11          | 1             | 1                          | 0.84          | 100             | 1.88          |
| 12          | 3             | 14                         | 16.73         | 100             | 3.93          |
| 13          | 5             | 16                         | 34.85         | 85              | 13.63         |
| 14          | 7             | 15                         | 23.83         | 100             | 3.00          |
| 15          | 4             | 11                         | 34.42         | 97              | 5.19          |
| 16          | 6             | 20                         | 46.19         | 94              | 6.29          |
| 17          | 1             | 35                         | 10.57         | 100             | 2.64          |
| 18          | 3             | 79                         | 3.56          | 97              | 1.51          |
| 19          | 8             | 32                         | 10.32         | 98              | 1.83          |
| 20          | 5             | 19                         | 10.45         | 98              | 2.08          |
| 21          | 1             | 28                         | 19.88         | 97              | 5.83          |
| 22          | 6             | 6                          | 36.37         | 100             | 3.51          |
| <b>Avg.</b> |               |                            | <b>18.76</b>  |                 | <b>3.47</b>   |

Table 2.3.  
A Comparison of Two Methods for Dataset 2, No-motion videos.

| Test        | Skin<br>Tones | Picard's approach [28, 29] |               | Proposed Method |               |
|-------------|---------------|----------------------------|---------------|-----------------|---------------|
|             |               | AccRate [%]                | $\mu_E$ [bpm] | AccRate [%]     | $\mu_E$ [bpm] |
| 1           | 2             | 12                         | 12.80         | 97              | 5.68          |
| 2           | 8             | 17                         | 17.26         | 96              | 6.90          |
| 3           | 3             | 24                         | 12.54         | 100             | 4.50          |
| 4           | 9             | 24                         | 27.96         | 97              | 3.37          |
| 5           | 6             | 23                         | 18.37         | 98              | 4.13          |
| 6           | 7             | 23                         | 46.47         | 97              | 3.70          |
| 7           | 7             | 16                         | 20.72         | 98              | 2.85          |
| 8           | 8             | 13                         | 25.69         | 97              | 5.62          |
| 9           | 4             | 34                         | 7.86          | 100             | 5.34          |
| 10          | 7             | 18                         | 13.48         | 100             | 3.39          |
| 11          | 10            | 27                         | 24.19         | 24              | 17.74         |
| 12          | 3             | 20                         | 20.55         | 100             | 3.21          |
| 13          | 1             | 24                         | 22.58         | 41              | 11.16         |
| 14          | 3             | 12                         | 18.35         | 100             | 3.83          |
| 15          | 3             | 38                         | 5.41          | 100             | 4.14          |
| 16          | 2             | 44                         | 7.02          | 100             | 3.36          |
| 17          | 1             | 26                         | 31.66         | 96              | 3.21          |
| 18          | 9             | 17                         | 14.99         | 97              | 1.62          |
| <b>Avg.</b> |               |                            | <b>19.33</b>  |                 | <b>5.21</b>   |

Table 2.4.  
A Comparison of Two Methods for Dataset 2, Non-random motion videos.

| Test        | Skin<br>Tones | Picard's approach [28, 29] |               | Proposed Method |               |
|-------------|---------------|----------------------------|---------------|-----------------|---------------|
|             |               | AccRate [%]                | $\mu_E$ [bpm] | AccRate [%]     | $\mu_E$ [bpm] |
| 1           | 2             | 7                          | 34.30         | 95              | 3.96          |
| 2           | 8             | 31                         | 18.20         | 95              | 10.26         |
| 3           | 3             | 20                         | 9.07          | 73              | 14.95         |
| 4           | 9             | 16                         | 11.87         | 96              | 26.84         |
| 5           | 6             | 9                          | 20.75         | 100             | 25.14         |
| 6           | 7             | 15                         | 19.85         | 95              | 15.96         |
| 7           | 7             | 20                         | 18.21         | 85              | 49.79         |
| 8           | 8             | 23                         | 14.34         | 100             | 8.77          |
| 9           | 4             | 34                         | 13.70         | 86              | 18.70         |
| 10          | 7             | 20                         | 22.87         | 100             | 3.32          |
| 11          | 10            | 14                         | 19.51         | 87              | 39.26         |
| 12          | 3             | 32                         | 13.47         | 100             | 2.44          |
| 13          | 1             | 31                         | 17.02         | 97              | 25.85         |
| 14          | 3             | 17                         | 11.90         | 94              | 7.15          |
| 15          | 3             | 23                         | 21.67         | 100             | 11.44         |
| 16          | 2             | 37                         | 12.88         | 100             | 30.36         |
| 17          | 1             | 17                         | 13.04         | 100             | 3.35          |
| 18          | 9             | 29                         | 16.50         | 100             | 9.62          |
| <b>Avg.</b> |               |                            | <b>17.17</b>  |                 | <b>17.07</b>  |

### 3. UNDERSTANDING MOTION EFFECTS IN VIDEOPLETHYSMOGRAPHY (VHR)

#### 3.1 Motion and Illumination in VHR

Our system described in Chapter 2 assumes that the RGB trace, the average intensity of RGB channels over time, composed of linear mixtures of PPG signal and other unknown noises. This assumption on linearity fails when there is subject motions in the video. In this Chapter, we investigate the relationship between the motion and the corresponding traces acquired from the video to understand the motion effects.

One of the major cause of the pixel intensity changes when there is motion is the change of the illumination  $I$  on the skin surface caused by motion. Moco *et al.* [61] provided experiments showing that orthogonal illumination minimizes the motion artifact in video-based PPG. For a single point light source, we can obtain the image intensity  $L$  in terms of the incident angle  $\theta$ , illumination  $I$ , and reflectance  $R$  of the surface where  $\theta$  is the angle between the incident light and the surface normal [133].

$$\begin{aligned} L &= IR \\ I &= I_0 + I_s \cdot \cos\theta \end{aligned} \tag{3.1}$$

where  $I_0$  is the uniform diffuse illumination,  $I_s$  is the illumination from a point source. In case of video pixel intensity  $L(n)$  where  $n$  is the frame index, Equation 3.1 can be rewritten as

$$L(n) = I_s R(n) \cos[\theta(n)] + I_0 R(n) \tag{3.2}$$

where  $\theta(n)$  would be the motion-related term and  $R(n)$  is a linear mixture of PPG signal  $h(n)$  and other signals [35, 57].  $\theta(n)$  in Equation 3.2 would be constant over frames when there is no motion. For this no motion case,  $L(n)$  is approximately the linear mixture of  $h(n)$  and other signals that we can use ICA and linear filters to



recover the underlying source signals as in our system in Chapter 2. When there is motion,  $L(n)$  is no longer a linear mixture of  $h(n)$  and noises but it includes the multiplicative motion term  $R(n)\cos[\theta(n)]$ .

Several recent VHR papers address the motion effects with multiplicative models. Feng *et al.* [54] describes a multiplicative motion model for video intensities  $L(n)$  in terms of PPG signal  $h(n)$ . They claim that when the subject is moving, the motion will modulate all three PPG signals in the RGB channels in the same way, as

$$L(n) = \alpha\beta(\gamma\mathbf{S}_0 \cdot \mathbf{h}(\mathbf{n}) + \mathbf{S}_0 + \mathbf{R}_0)M(n) \quad (3.3)$$

where  $M(n)$  is the motion modulation,  $\alpha$  is the power of the light in the normalized practical illumination spectrum (corresponding to  $I$  defined in Equation 3.1),  $\beta$  is the power of the light in the normalized diffuse reflection spectrum of the skin,  $\gamma$  is the ac/dc ratio of PPG signal,  $S_0$  is the average scattered light intensity from skin and  $R_0$  is the diffuse reflection light intensity from the surface of the skin. Kumar *et al.* [35] described  $L(n)$  as the multiplicative model of the intensity of illumination  $I$  and the reflectance of the skin surface  $R$ . Combining this with the camera noise  $q(n)$ , they proposed the following model.

$$L(n) = I(\mathbf{a} \cdot \mathbf{h}(\mathbf{n}) + \mathbf{b}) + q(n) \quad (3.4)$$

where  $a$  is the strength of blood perfusion, and  $b$  is the surface reflectance from the skin. They addressed that change in  $I$  can corrupt the PPG estimate and small light direction changes caused by motion can lead to large changes in skin surface reflectance  $b$ . Haan *et al.* [57] proposed a similar model

$$L(n) = I(\mathbf{c} + \mathbf{h}(\mathbf{n})) \quad (3.5)$$

where  $c$  is the stationary part of the reflection coefficient of the skin. Their recent work [58–60] specifically address solutions to motion problems.

While Equations 3.3, 3.4 and 3.5 have different approaches and notations in modeling  $L(n)$ , they all assumes that  $L(n)$  includes the multiplication between the illumination and reflection. And in all three models, the terms for reflectance  $R(n)$ ,

denoted in boldface in each equation, are represented as the linear mixture of  $h(n)$  and other (constant or varying) terms. In this thesis, we take this idea that  $R(n)$  is a linear mixture of  $h(n)$  and other signals where the other signals would not change over time in short video scripts for a specific skin surface point.

### 3.2 Simple Modeling: Intensity Change of Moving Object

In this section, we analyze how motions affect intensity of a specific object surface point in moving object video. For a video taken from a camera with a single light source where the positions of both the camera and the light source are fixed, the motions inside the video play a significant role in the pixel intensity changes in the video. This is because each surface point has a corresponding incident angle  $\theta$  and motion in the video changes  $\theta$  throughout frames as described in Equation 3.1.

These intensity changes caused by motion are often times small and barely noticeable in human eyes. In VHR, the PPG signal that reflects heart beat is even smaller and is not even noticeable in human eyes. In our no-motion dataset described in Chapter 2, the average intensity variation of green channel within the face skin region for 10-second duration was approximately 2% on average ( $L_{min}/L_{max} = 0.98$  on average where  $L_{min}$  and  $L_{max}$  are minimum and maximum intensities of green channel respectively). Figure 3.1 shows an example of the average intensity of green channel, we call green trace, within face skin region for 10-second duration. These small variations in the average green trace would contain PPG signal together with all the other noises. The motion-related signals have severe effect on VHR and makes it difficult to obtain the HR related signal in the video.

In order to see how much intensity change is caused by motions, we assume a simple motion model in a constrained shooting environment. Let's assume we are observing the intensity at a specific point of a sphere in a video. Figure 3.2 shows the top view of this shooting environment with light rays falling on specific points of a sphere. The sphere only moves in left and right (LR) directions and there is a one

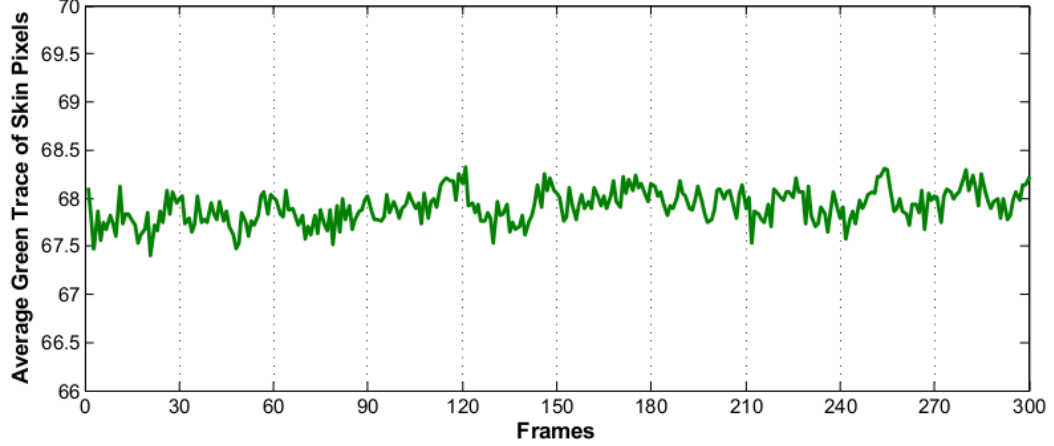


Fig. 3.1. Average green trace within face skin region for 10-second duration for subject 17, Dataset 1. The range of intensity  $L$  for all color channels in Dataset 1 is:  $L \in [0, 255]$ . The average HR obtained from pulse oximeter for this 10-second duration is 64 bpm (meaning about 10.7 beats for 10 second).  $L_{min} = 67.4$ ,  $L_{max} = 68.3$  and  $L_{min}/L_{max} = 98.7\%$ .

point light source with a fixed location. The sphere is an approximate modeling of a human head. Camera viewing the head is not shown in Figure 3.2 and it is assumed to be somewhere between the head and the light source.

For the center point of the head denoted with blue points in Figure 3.2-(a), the intensity  $L$  of the center point reaches maximum when  $\theta = 0$  ( $d = 0$ ) following from Equation 3.1. This is when the surface point is right in front of the light source.  $L$  decreases as the head moves away from the center. In this restricted motion scenario, we can obtain the minimum and maximum intensities of a single surface point based on Equation 3.1 where  $I_0 = 0$  assuming that there is only a point light source. Equation 3.6 shows  $L_{max}$  and  $L_{min}$  for center point ( $\beta = 0^\circ$ ).

$$\begin{aligned}
 L_{max} &= I_s R \\
 L_{min} &= L_{max} \cos[|\theta|_{max}] \\
 |\theta|_{max} &= \tan^{-1} \frac{d}{D-r}
 \end{aligned} \tag{3.6}$$

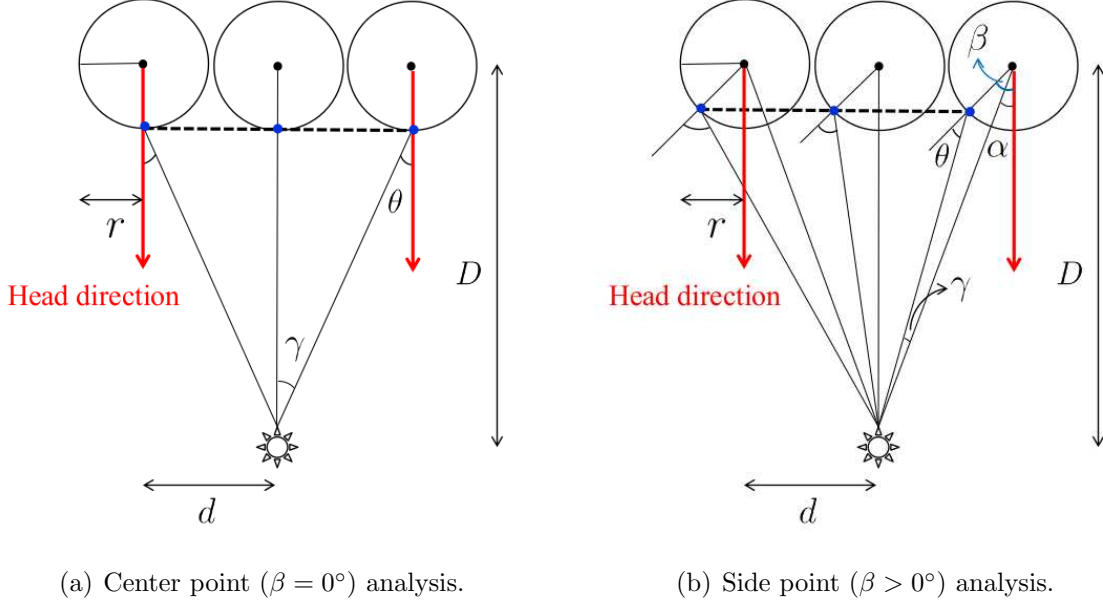


Fig. 3.2. Point analysis for a simple motion model.  $\theta$  is the incident angle,  $r$  is the radius of the head when viewed from the top,  $d$  is the moved distance,  $D$  is the distance between the source light and the line of movement,  $\alpha$  is the angle from the head direction to the line connecting center of head and the light,  $\beta$  is the angle between specific face point and head direction from the center of the head, and  $\gamma$  is the angle between specific face point and the center of head from the light source.  $\alpha$  is zero when the head direction is toward the light and aligned with the line between the source light and the center of the head.  $\alpha$  is positive when in counterclock-wise direction.  $\gamma$  is zero when the face point is on the line between the light source and the center of the head.  $\gamma$  is positive when in counterclock-wise direction. In both figure(a) and (b), the leftmost circle denotes the farthest position to the left and the rightmost circle denotes the farthest position to the right.

Equation 3.6 can be extended to a side point case ( $\beta = 0^\circ$ ) by introducing three additional variables  $\alpha$ ,  $\beta$  and  $\gamma$  as denoted in Figure 3.2-(b).

$$\theta = \beta - \alpha + \gamma \quad (3.7)$$

where  $d$  and  $\alpha \in [-\tan^{-1}(|d|_{max}/D), \tan^{-1}(|d|_{max}/D)]$  is a motion related variable,  $\beta$  is an angle that denotes the specific point of a face.  $\beta$  is constant for each point on a face. When  $\beta \neq \alpha$ , the value for  $\gamma$  satisfies the following equation.

$$\sqrt{\frac{1}{\sin^2 \gamma} - 1} = \frac{\sqrt{(D/r)^2 + (d/r)^2} - \cos(\beta - \alpha)}{|\sin(\beta - \alpha)|} \quad (3.8)$$

From eq. 3.7 and eq. 3.8, we can obtain the corresponding  $\cos \theta$  for a moved distance  $d$ . This shows the relation between the intensity  $L_{max} \cos \theta$  and a moved distance  $d$ .

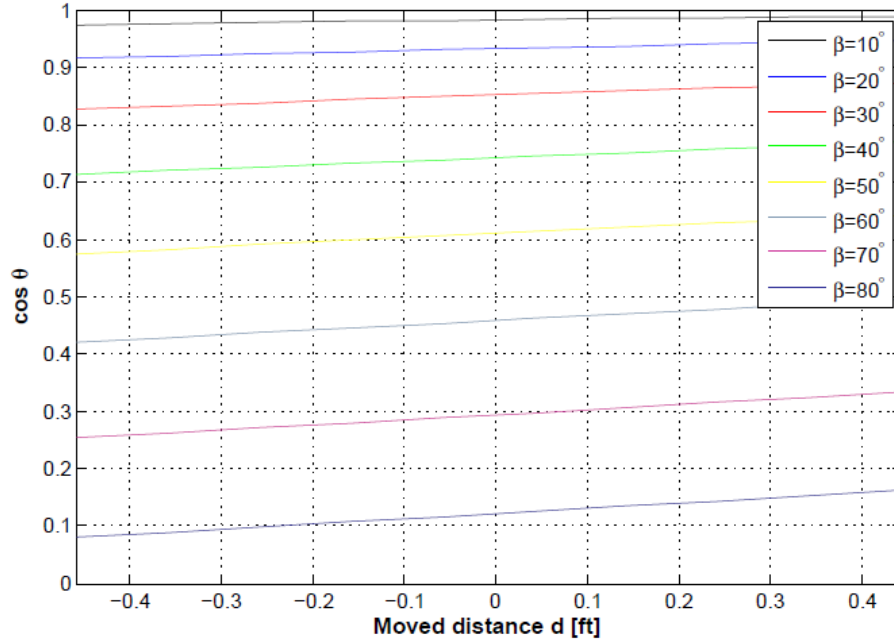


Fig. 3.3. Relation between moved distance  $d$  and  $\cos \theta$  for various  $\beta > 0$ .

If the angle  $\beta$  is small (the point is close to the center of the face), there is not much intensity change for  $d$  within  $[-0.46, 0.46]$  [ft] range. For large  $\beta$ , the  $L_{min}/L_{max}$  ratio increases. As shown in Table 3.1, for a facial point at the side of the face with  $\beta = 80^\circ$ , the ratio  $L_{min}/L_{max}$  drops to 0.495.

We made test videos to see if we can observe this relationship between  $\beta$  and  $L_{min}/L_{max}$  ratio. Figure 3.4 shows the data collection environment. We tried to mimic the simple modeling in Figure 3.2 but used the flat surface box instead of the sphere in order to reliably obtain the intensity  $L(n)$  of a specific surface plane. We

Table 3.1.  
 $r_L = L_{min}/L_{max}$  when  $|2d| \leq 11/12$ ,  $D = 11$ , and  $r = 7/12$ .

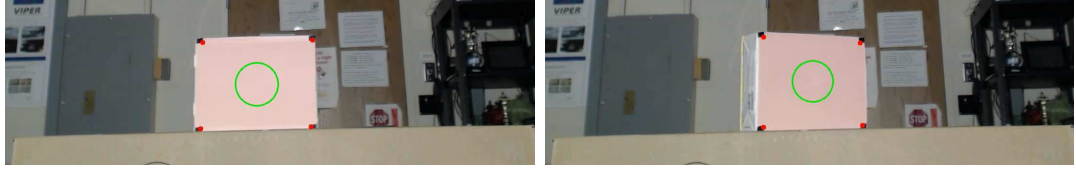
| $\beta$ [°] | $r_L$ |
|-------------|-------|
| 0           | 0.999 |
| 10          | 0.984 |
| 20          | 0.967 |
| 30          | 0.949 |
| 40          | 0.926 |
| 50          | 0.896 |
| 60          | 0.850 |
| 70          | 0.762 |
| 80          | 0.495 |



Fig. 3.4. The data collection environment. The distance  $D$  between the object's moving plane and the light is 11 ft. The range of moving distance  $d$  along the moving plane is  $|2d| < 11$  inch. The height of the light  $h_l$  and height of the object  $h_o$  are similar ( $h_l = 47$  and  $h_o = 43$  inches). The object surface facing the camera is a paper in solid color of light pink.

shoot the videos with Logitech Webcam c920 in lossless format with both the auto white balance and auto gain control options set to OFF. The videos were  $1920 \times 1080$  in 30 fps. The room had no windows and all the room lights on the ceilings were off when the videos were taken. Only one light source shown in Figure 3.4 was turned on. While keeping all the other conditions the same, we made three different surface

angles to make incident angles of 0, 30 and 60 degrees. The object moved through the same moving plane which is perpendicular to the incident light ray. A researcher manually moved the object and maintained the surface angle of the object by fixing the object on a paper where the protractor is printed on. Each videos were 50-second length (25-second length for No-motion and the other 25-second for Motion). Figure 3.5 shows the video captures for three different test cases.



(a)  $\beta = 0^\circ$ . The number of ROI pixels was 6077 for all frames. (b)  $\beta = 30^\circ$ . The number of ROI pixels was 5525 for all frames.



(c)  $\beta = 60^\circ$ . The number of ROI pixels was 4053 for all frames.

Fig. 3.5. Camera views of test videos in different angles. The average  $L(n)$  is obtained from the ROI pixels within the green circle—the radius of the circle is the diagonal distance between two red points divided by 6.5. Four corner points in red are manually selected in the first frame and obtained by feature tracker [134] in the rest of the frames.

Figure 3.6 shows the Average  $L$  of ROI in R channel. For all test cases, there is a notable difference between No-motion and Motion in terms of average  $L$ . While the average  $L$  does not change throughout time for No-motion case, the average  $L$  varies in Motion resulting up to  $r_L = L_{min}/L_{max} = 0.937$  for  $\beta = 60^\circ$ . In this scenario, the light source is approximately a point source and the  $R(n) = R$  is constant over time. Therefore, Equation 3.2 can be simplified to

$$L(n) = I_s R \cos[\theta(n)]. \quad (3.9)$$

This means that the average  $L(n)$  shown in Figure 3.6 is the  $\cos[\theta(n)]$  scaled by  $I_s R$ .

Table 3.2 shows  $r_L$  obtained from each RGB channel along with the “Simulated  $r_L$ ” shown in Table 3.1. The average  $r_L$  obtained from RGB traces does not exactly

Table 3.2.  
 $r_L = L_{min}/L_{max}$  [%].

| $\beta$ [°] | Simulated<br>$r_L$ | $r_L$<br>in B ch. | $r_L$<br>in G ch. | $r_L$<br>in R ch. | Average<br>$r_L$ | Std. of<br>$r_L$ |
|-------------|--------------------|-------------------|-------------------|-------------------|------------------|------------------|
| 0           | <b>99.9</b>        | 97.82             | 98.18             | 98.59             | <b>98.20</b>     | 0.4              |
| 30          | <b>94.9</b>        | 95.23             | 93.80             | 94.94             | <b>94.66</b>     | 0.8              |
| 60          | <b>85.0</b>        | 95.20             | 93.42             | 92.38             | <b>93.66</b>     | 1.4              |

match to the simulated  $r_L$  but the tendency that for higher  $\beta$ ,  $r_L$  gets lower holds for both simulated  $r_L$  and RGB trace-based  $r_L$ . We have not considered the camera quantization or other camera processes and this could be the reason for the mismatch between two different  $r_L$ .

In conclusion, the motion effects vary a lot for different point of the face due to their surface angle differences. Most of current VHR methods begin with taking an average  $L(n)$  of each RGB channel over the entire face/face skin/sub-region of face (cheek, forehead) in order to obtain  $h(n)$ . As observed from our experiment, motion-generated signal  $\theta(n)$  for each different point on face could be completely different signals depending on what kind of motion there is. The trace obtained by taking the average over multiple surface points with various surface angles will result in both non-linear and linear mixtures of  $h(n)$  and other noises including motion-related signal.



### 3.3 Intensity Model in Human Video with Motion

In section 3.2, we considered a constrained model where only LR movements are possible and the head of a complete sphere always maintains the same direction. In real human video, LR movements involve head rotations in up/down or left/right directions as well. Those variations make changes to the incident angles. In this section, we obtained an incident angle of a specific facial point throughout frames to see the motion effects described in section 3.2 in real human videos.

By introducing the surface normal to Equation 3.2, the image intensity in terms of the surface normal, illumination, and reflectance of the surface can be rewritten as follows.

$$L_k(n) = R_k(n) \left[ \sum_j I_{jk} \cdot \vec{l}_{jk}(n) \cdot \vec{n}_k(n) + I_0 \right] \quad (3.10)$$

$k$  is an index for specific point on facial skin,  $j$  is an index denoting each point light source,  $I_{jk}$  is the amplitude of the light from light source  $j$  to the skin surface point  $k$ ,  $\vec{l}_{jk}(n)$  is the unit vector for the light ray from point  $k$  to light source  $j$ ,  $\vec{n}_k(n)$  is the unit vector for the surface normal at skin surface point  $k$ ,  $R_k(n)$  is the reflectance at skin surface point  $k$ , and  $n$  is an index for frame number.

For point light sources coming from the same lighting, we can have approximate light ray on point  $k$ .

$$\sum_j I_{jk} \cdot \vec{l}_{jk}(n) \approx I_k \cdot \vec{l}_k(n) \quad (3.11)$$

$$L_k(n) = R_k(n) I_k(n) \quad (3.12)$$

$$I_k(n) = I_k \cdot \vec{l}_k(n) \cdot \vec{n}_k(n) \quad (3.13)$$

$L_k(n)$  is what we can observed from video,  $R_k(n)$  is the reflectance that contains HR signal,  $I_k(n)$  is the illumination that varies with incident angle.

$$L_k(n) = R_k(n) \left[ I_k \vec{l}_k(n) \cdot \vec{n}_k(n) + I_0 \right] \quad (3.14)$$

Equation 3.14 still involves five different unknown variables or constants. By letting  $\vec{n}_k(n) = \vec{n}(n) + \vec{d}_k(n)$  where  $\vec{n}(n)$  is face direction normal to the arbitrary global face plane and  $\vec{d}_k(n)$  is a vector denoting the difference between  $\vec{n}_k(n)$  and  $\vec{n}(n)$ , we can have further approximations. For those skin surface points where the surface angle relative to the face direction is almost fixed—the skin surface where the facial muscle movements are negligible,  $\vec{d}_k(n) \approx \vec{d}_k$  and if the distance between the light and the head is much longer than the head movements,  $I_k \cdot \vec{l}_k(n) \approx I \cdot \vec{l}$ .

$$L_k(n) = R_k(n) \left[ \vec{n}(n) \cdot I\vec{l} + \vec{d}_k \cdot I\vec{l} + I_0 \right] \quad (3.15)$$

In Equation 3.15, the first term is not related to the specific skin surface—it is a common term related to the head movements. The second and the third term are constants that does not involve the frame index  $n$  and can be replaced with the constant  $c_k$ .

$$L_k(n) = R_k(n) \left[ \vec{n}(n) \cdot I\vec{l} + c_k \right] \quad (3.16)$$

If we let  $R_k(n) \approx a_k h(n) + b_k$  where  $a_k$  is the strength of blood perfusion,  $b_k$  is the surface reflectance from the  $k$ th skin point [35] and  $h(n)$  is the PPG signal from the heart beat, we have

$$L_k(n) = a_k h(n) \vec{n}(n) \cdot I\vec{l} + b_k \vec{n}(n) \cdot I\vec{l} + a_k c_k h(n) + b_k c_k. \quad (3.17)$$

Equation 3.17 shows that the intensity change on skin point  $k$  observed from the video is linear mixture of four different terms. What this model means is that if there is no head motion,  $L_k(n)$  would be the linear mixture of  $h(n)$  and constants not depending on  $n$ . If there are head motions, it is difficult to directly observe the signal  $h(n)$  through  $L_k(n)$ . For periodic head motions,  $m_h(n) = \vec{n}(n) \cdot I\vec{l}$ , the first term in Equation 3.17 is modulation of two periodic signals  $h(n)$  and  $m_h(n)$ .

$$L_k(n) = a_k h(n) m_h(n) + b_k m_h(n) + a_k c_k h(n) + b_k c_k. \quad (3.18)$$

The modulation effect will cause several peaks in the PSD of  $L_k(n)$ .

### 3.4 Filters

Adaptive filter can be used to remove the motion artifact if we have a reference signal that has the strong correlation to the motion artifact but uncorrelated to the PPG signal. Huang *et al.* [71] described the signal of skin color changes as three components of the blood volume variation, human motion and the ambient light change. Along with RGB traces of skin pixels within the face region, they also obtained the trace of  $(x, y)$  coordinates of the ROI and used them as inputs to cascade adaptive filter to alleviate the interference related to motion [71]. They experiment with one long video of only one subject during exercising on treadmill. Their assumption (linear mixtures) on motion effects is different from ours (multiplicative effect) but the idea of using the motion information,  $(x, y)$  coordinates in their case, as a reference signal in adaptive filter to reduce the motion effects in the RGB traces is the same with what we are going to pursue next.

Based on our model in Equation 3.18, homomorphic filter [135, 136] might be used together with other linear filters to reduce the motion artifact. Homomorphic filter is used for signals combined in a nonlinear way. It first transforms nonlinearly related signals to additive signals. Then, the signal is processed by linear filters such as bandpass filter (BPF) and it is transformed backward by the inverse nonlinearity. An example of transforming and inverse transforming nonlinearly related signals is taking the logarithm and exponential to the signal. Two multiplied signals become additive when logarithm is taken. Figure 3.7 shows an example of block diagram for using homomorphic filter in video-based HR measurement. This method is left for future work and it require both facial landmark detection and facial direction estimation accurately done on each frame of the video along with direction of the light source.

### 3.5 Region Selection and Face Direction

Given three points in 3D space, a unique plane is determined. The surface normal at point  $k$ ,  $\vec{n}_k(n)$ , can be obtained if we know three points on the surface that are not aligned on one line.

Human head is not a perfect sphere that it is difficult to know the skin surface normal for each point  $k$ . We can get an approximate surface normals through detecting three specific points that form a surface that we want. As what we want to obtain is the direction of the surface but not the absolute 3D location of the surface, we can use the relative three points to form the surface. First we set two points as the center of the left and right eyes. Then, we find the 2D point of the tip of the nose from the frame. In our experiment, the head movement is restricted to the image plane. And from our observations, we assume that the subject rotate the head to left direction when moving to the left side and rotate to right direction when moving to the right side. If two eye points are on the image plane, the tip of the nose point shown on the image is what is being projected from the 3D point to the image plane. We assume that the sign of  $z$  coordinate of the tip of the nose does not change throughout the video since the subject was asked to look toward the camera. The  $(x, y)$  coordinates of the three points can be determined from the tracking points in the image plane and the  $z$  coordinate for the tip of the nose can be determined by the length of the nose obtained from the image and the length of the nose obtained from the side view.

Let  $\vec{n}_k(n)$  be  $[n_x(n), n_y(n), n_z(n)]$ . Let  $M = [X, A, B, C]$  where  $X$  is a general point on a plane and  $A$ ,  $B$ , and  $C$  are three selected points of the face. Each  $X$  can be expressed as a linear combination of the other three that  $\det(M) = 0$  [137]. By

using this property, we can attain the plane normal  $\vec{n}_k(n)$  [84, 137]. Let  $\tilde{M}$  denote the  $4 \times 3$  submatrix formed by the known last three column vectors of  $M$ .

$$\tilde{M} = \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.19)$$

where  $X_1$ ,  $X_2$ , and  $X_3$  denote  $x$ ,  $y$  and  $z$  coordinates of point  $X$  respectively. Let  $D_{ijk}$  stand for the determinant obtained from the  $i_{th}$ , the  $j_{th}$ , and the  $k_{th}$  rows of  $\tilde{M}$ . Then, we have [84, 137]

$$\vec{n}_k(n) = \begin{bmatrix} D_{234} \\ -D_{134} \\ D_{124} \end{bmatrix}. \quad (3.20)$$

With these three estimated points on each frame and the approximate light source direction, we can estimate the motion-caused variation of  $L(n)$ . We denote the  $L(n)$  estimated from the tracking points as  $L(\hat{n})$ .  $L(\hat{n})$  would not involve PPG signal because it is estimated solely from illumination and motion without using pixel intensities.  $L(n)$  was obtained from the pixels within the circle where the center of the circle is center of the noise point obtained in every frame through feature tracking and the radius of the circle is set to 3 pixels (number of pixels inside the circle: 28). For the unit vector from ceiling light to the point  $k$  on the skin, we used  $\vec{l}_k(n) = [-0.2576, -0.8013, -0.5446]$  obtained from measurements in the shooting environment. This  $I_k$  would slightly vary as the person moves but we used the fixed values as an approximation. We assigned fixed values for unknown scaling factor  $R_k(n) \cdot I_k = C$  in a way such that the mean of Average  $L(n)$  and the mean of  $L(\hat{n})$  are equal.

$$L(\hat{n}) = C \cdot \vec{l}_k(n) \cdot \vec{n}_k(n) \quad (3.21)$$

Figure 3.8 shows an example of facial points and ROIs. Figure 3.9 shows the Average  $L(n)$  and  $L(\hat{n})$ . The  $L(\hat{n})$  is obtained from  $\vec{n}_k(n)$  estimated from three points A, B, C which were obtained from each frame of the video.

Limitations of our experiment is that we used approximations and assumptions such as  $I_0 = 0$ ,  $I_k$  is constant and  $\vec{l}_k(n)$  is fixed. And we used very small region for obtaining the trace which would not contain a reliable PPG signal due to its small size. Despite the limitations, Figure 3.9 shows that the Average  $L(n)$  and  $L(\hat{n})$  estimated only from the motion information are similar both in time-domain. The small variations (fluctuations) are only shown in  $L(n)$ . In the frequency domain, motion peaks appeared around 0.17 Hz in the PSD plots for both the Average  $L(n)$  and  $L(\hat{n})$  in both subjects (not shown in the plot).  $L(\hat{n})$  contains a periodic signal with frequency of 0.17 Hz and from Equation 3.21,  $I_k(n)$  also contains a periodic signal with frequency of 0.17 Hz.  $L_k(n)$  and  $L(\hat{n})$  share the same  $I_k(n)$ . From the fact that the peak around 0.17 Hz is observed both in  $I_k(n)$  and  $L_k(n)$ , we expect that  $R_k(n)$  would contain the constant term  $R_0$  that does not change over time. Equation 3.22 is an updated equation of Equation 3.12 rewritten to include constant term within  $R_k(n)$ .

$$L_k(n) = [R_h(n) + R_0]I_k(n) \quad (3.22)$$

where  $R_h(n)$  is a linear mixture of PPG signal  $h(n)$  and other reflectance terms.  $R_h(n)$  is modulated by  $I_k(n)$  unless  $I_k(n)$  is constant over time. As expected, no strong peak appeared around the GTHR within the frequency range of our interest (Figure 3.9). Our experiment shows strong relationship between  $I_k(n)$  and the cosine of the incident angle, estimated with  $\vec{l}_k(n) \cdot \vec{n}_k(n)$ . Further understanding on  $I_k(n)$  may compensate the modulation effect in  $L_k(n)$  for different facial point  $k$ .

### 3.6 Conclusion and Future Work

We showed the relationship between the motion and the intensity change by simple modeling for moving object with constant reflectance. We extended the experiment to human video and showed the motion effects on the intensity change in terms of the skin surface normal and illumination. Our results show how the incident angle change caused by motion is related to the pixel intensity changes. We showed that the

illumination change on each surface point is one of the major factors causing motion artifacts.

Following are suggested for future work: (1) estimating  $L(\hat{n})$  could be done more accurately by improving the tracking performance of three facial points, (2) instead of using fixed values for all the frames, the light source direction for each frame could be estimated using the location and shadow information and (3) a method to find sub-region (surface) that share the same surface normal ( $\vec{n}_k(n)$ ) could be investigated so that more surface normal estimation can be done more accurately.

There are several related work to note regarding future work.

Lin *et al.* [75] addressed that the success of face-based HR estimation strongly depends on the measurement of facial illumination. They suggested to have various assumptions and simplifications about illuminance and reflectance because separating the reflectance and the illuminance fields from real images is, in general, a poorly-posed problem. Their assumption was not clearly denoted in the paper, but they used  $x$  and  $y$  direction edge filters to give different weighting at each pixel. We also think that there should be different compensations for different regions since the illumination variations on each pixel differs due to different surface angles.

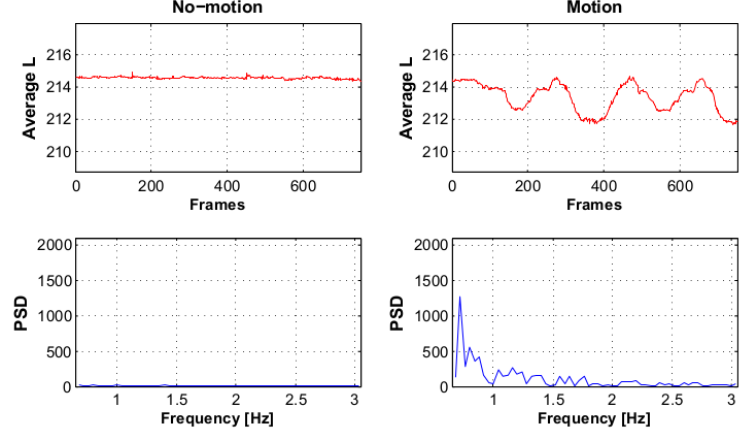
Kumar *et al.* [35] divided the facial regions into smaller regions do the tracking separately from the assumption that different part of the face contain different strength in of the PPG. The idea of having sub-regions makes sense because the illumination change on different sub-regions can be different. But the same sized grid of  $20 \times 20$  pixel blocks used in their paper does not necessarily mean the pixels within each block would share the same intensity change.

In Section 3.5, we manually selected the facial points in the first frame and used feature tracker [134] in the rest of the frames. This works for short period of time but the tracking error is accumulated as the duration of the video gets longer. Facial point detection methods for single image are described in [138, 139]. Several work on finding facial landmarks in video frames both for detection and tracking are described

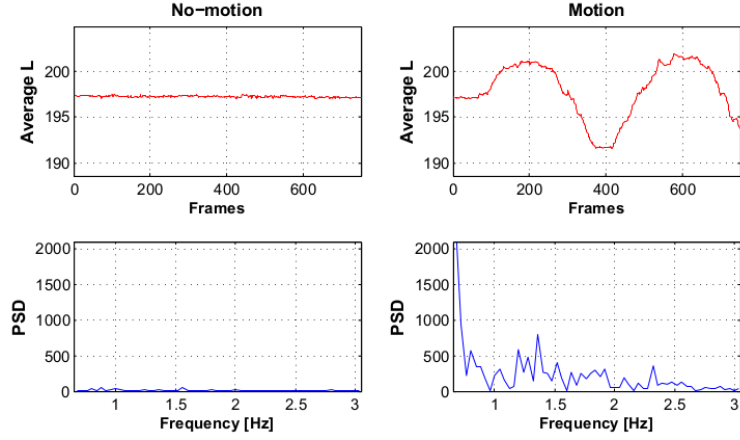
in some recent work [140–142]. facial landmark localization techniques should be further investigated.

In Section 3.5, we only experimented on videos with limited motion patterns. We were able to estimate the direction of the surface formed from three specific points (the centers of two eyes and the tip of the nose) just by estimating the distance between the eyes and the distance of the nose because of many restrictions in the motion patterns. To extend our approach to random motions, methods for estimating the face directions in the video [143–147] can be further investigated.

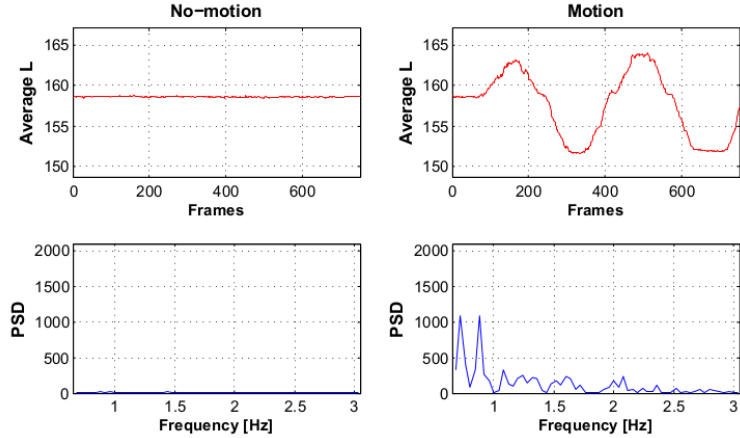




(a)  $\beta = 0^\circ$ .  $L_{max} = 214.7$ ,  $L_{min} = 211.7$  and  $r_L = 98.6\%$ .



(b)  $\beta = 30^\circ$ .  $L_{max} = 201.8$ ,  $L_{min} = 191.6$  and  $r_L = 94.9\%$ .



(c)  $\beta = 60^\circ$ .  $L_{max} = 164.1$ ,  $L_{min} = 151.6$  and  $r_L = 92.4\%$ .

Fig. 3.6. Average  $L$  of ROI in R channel in three different surface angles: No-motion vs. Motion.  $L$  is 8bits/pixel/channel and  $L \in [0, 255]$ . The PSD of each trace (the average  $L(n)$ ) within the frequency range of our interest in VHR,  $f_l = 0.7$  and  $f_h = 3.0$  Hz, is plotted in blue below the each trace.

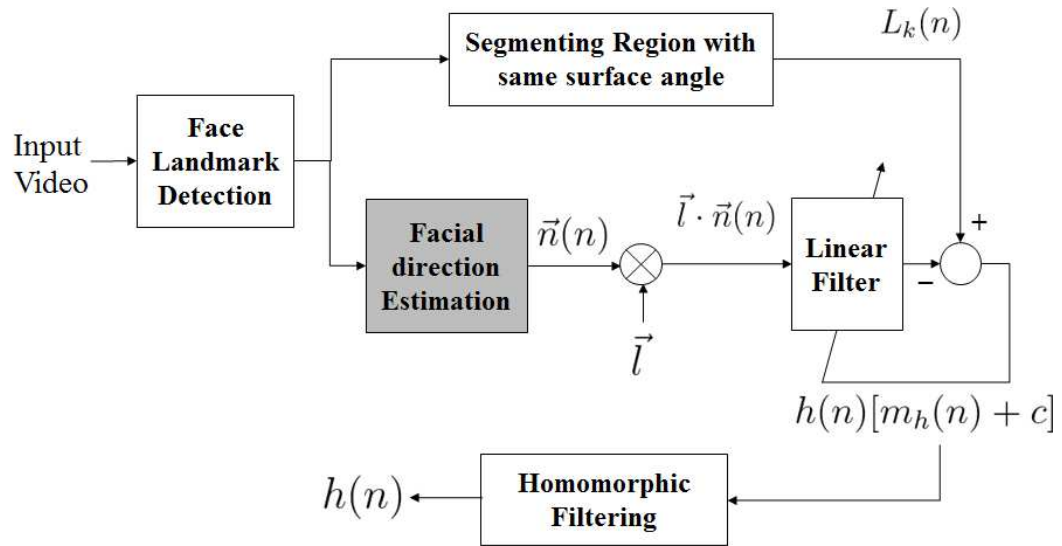


Fig. 3.7. Block Diagram.

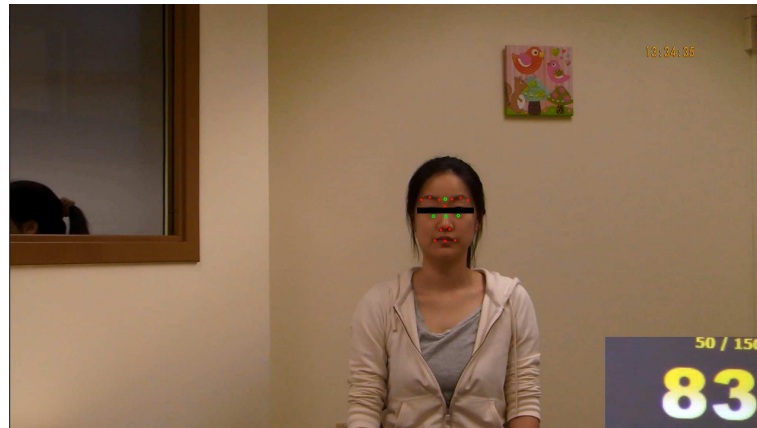
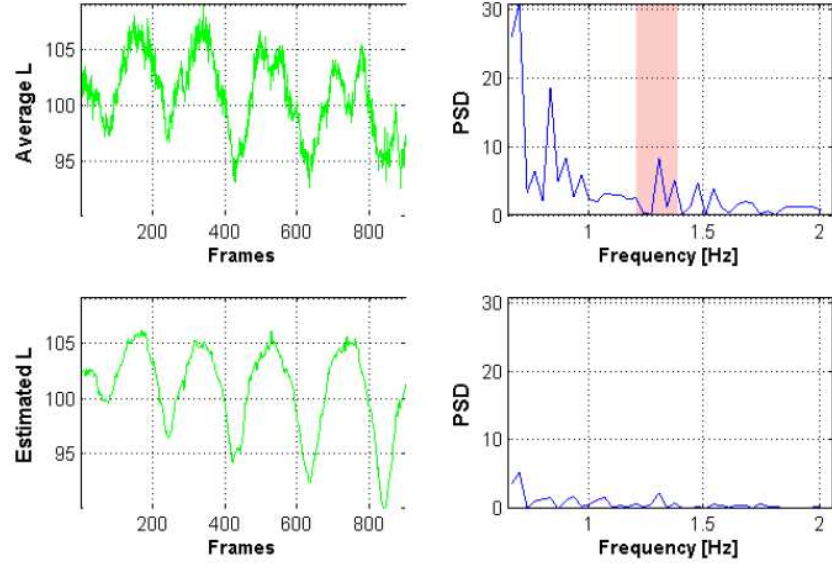
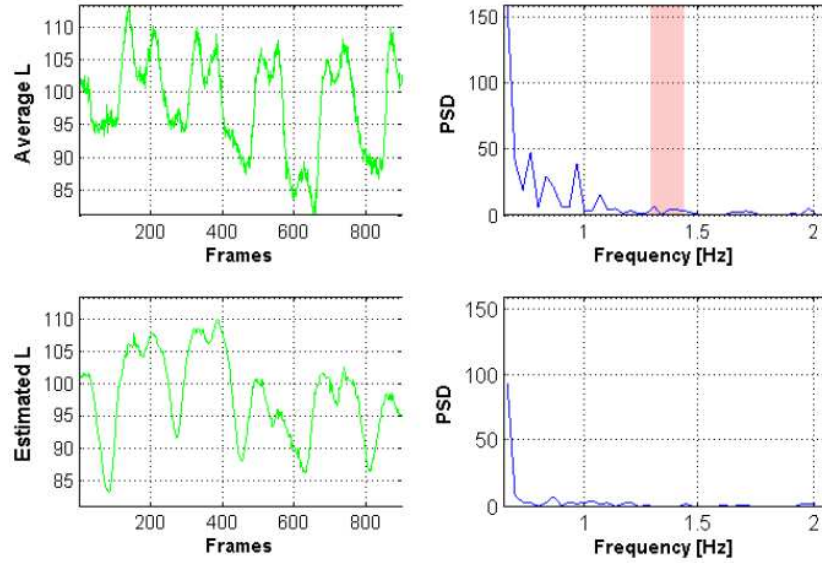


Fig. 3.8. An example captured from Dataset 2. Facial points denoted in red. ROI regions denoted in green—only the ROI in the middle of the nose was used.



(a) Subject 3



(b) Subject 14

Fig. 3.9. Experimental result on Dataset 2 of non-random motion videos: Average  $L(n)$  and  $\hat{L}(n)$ .  $\hat{L}(n)$  is denoted as “Estimated L” in the plot label. The red patches in PSD plots denote the GTHR range for each subject. The frequency range in PSD plot is  $f_l = 0.7$  and  $f_h = 2.0$  Hz. Both subject 3 and 14 showed strong peak around 0.17 Hz corresponding to motion (Not shown on the plot).

## 4. SLEEP ANALYSIS USING MOTION AND HEAD DETECTION

Pediatric sleep medicine is a field that focuses on typical and atypical sleep patterns in children. Within this field, physicians, interventionist, and researchers record and label child sleep with particular attention to sleep onset time, total sleep duration, and the presence or absence of night awakenings. One notable recording method is videosomnography (VSG) which includes the labeling of sleep from video [2, 3]. This method is most commonly used for infants/toddlers as their compliance rates with other sleep recording methods can be low. Traditional behavioral videosomnography (B-VSG) labeling includes manual labeling of awake and sleep states by trained technicians/researchers. B-VSG is time consuming and requires extensive training which has limited its widespread use within the pediatric sleep medicine field. Within the present study we develop and test an automated VSG method (auto-VSG) to replace B-VSG and to provide physicians, interventionist, and researchers with a sleep recording tool that is more economic and efficient than B-VSG, while maintaining high levels of labeling precision.

The development of auto-VSG is a growing area with preliminary studies utilizing signal processing systems that index movement during sleep in small groups of children with developmental concerns or adults [2, 93–95]. Across these studies, motion within the video is estimated by frame differencing [93, 94] or by obtaining motion vectors [2, 95]. However, each of these studies were completed within a controlled setting and do not account for the wide range of camera positions and lighting variations that are common among in-home VSG recordings. Within the present study, the proposed system adjusts for these ‘in the wild’ factors and uses two sleep field stands as comparison measures of sleep. The first is actigraphy, which estimates sleep

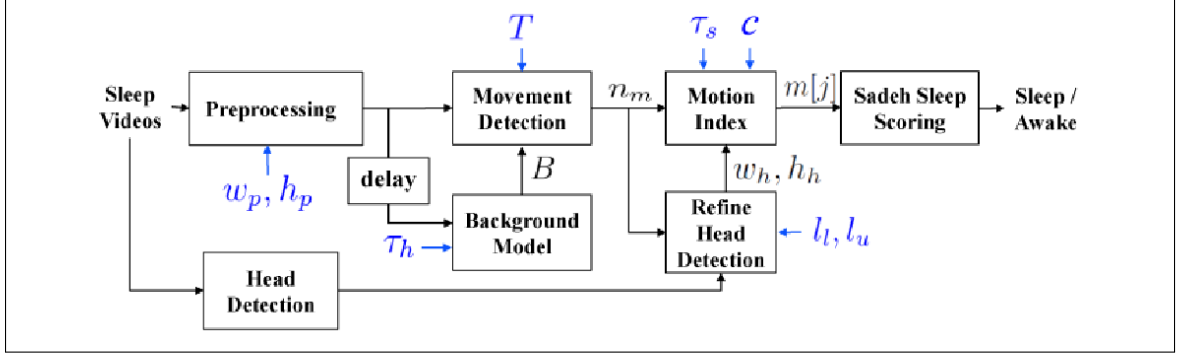


Fig. 4.1. Proposed Sleep Detection System.

vs. awake states based on child movement as indexed by an ankle worn accelerometer. Second, trained technicians/researchers provided sleep vs. awake estimates using traditional B-VSG labeling methods.

In this chapter, we develop and test an auto-VSG method that includes (1) preprocessing the video frames using histogram equalization and resizing, (2) detecting infant movements using background subtraction, (3) estimating the size of the infant by detecting their heads based on deep learning methods, and (4) scaling and limiting the degree of motion based on a reference size so the motion can be normalized to the size of the relative child in the frame. The generated estimates are then categorized as awake or sleep for each minute of video by applying an established sleep field algorithm [148]. Finally, all auto-VSG estimates were compared with those provided by actigraphy and B-VSG.

## 4.1 Sleep Detection

### 4.1.1 Motion Detection

We assume that there is less motion during sleep than awake states [149] and that the child is the only source of motion in the video. Background subtraction is widely used for detecting moving objects from static cameras [150]. Moving objects (foreground objects) are detected by taking the difference (subtraction) between the

background model and the current frame. While some background subtraction methods aim to detect moving objects as foregrounds, our system aims to detect moving regions as foreground. As shown in Figure 4.1, the system begins by converting the RGB video frame to a gray scale frame and resizing it to  $w_p \times h_p$  where  $w_p$  and  $h_p$  are width and height of the resized frame. Preprocessing includes histogram equalization to enhance gray scale contrasts. This helps adjust the overall image intensity range across various room lighting schemes. Next the background model is obtained from history of  $h[i]$  previous frames as in (4.1)

$$B_i[x, y] = \frac{1}{h[i]} \sum_{k=i-h}^{i-1} I_k[x, y] \quad (4.1)$$

where  $i$  is the video frame index,  $h[i] = \tau_h \cdot f_s$  is the number of previous frames (history) used for the background model in frame  $i$ ,  $\tau_h$  is the history in seconds,  $f_s$  is the frame rate of the video,  $I_i[x, y]$  is a pixel in frame  $i$  and  $B_i[x, y]$  is a pixel in the background model at frame  $i$ . The difference between the background model  $B_i[x, y]$  and  $I_i[x, y]$  indicates whether each pixel in the frame is classified as “moved” or “not moved”. A pixel is classified as “moved” if (5.1) holds.

$$|I_i[x, y] - B_i[x, y]| > T \quad (4.2)$$

where  $T$  is a threshold for determining movement for one pixel and for our experiments the value of  $T$  is empirically determined. We quantify the amount of movement as the number of pixels classified as “moved.” Note that if the history  $h[i]$  in (4.1) is set to a small value such as 1, the background model  $B_i$  would be almost identical to the current frame  $I_i$  and that the system will not properly detect the motion. We obtain the average number of moved pixels for time segment  $j$  as

$$m_j = \frac{1}{K} \sum_{k=0}^{K-1} n_m[k] \quad (4.3)$$

where  $k$  is frame index within one time segment,  $n_m[k]$  is number of moved pixels in frame  $k$ ,  $K$  is number of frames for one time segment,  $\lfloor \tau_s \cdot f_s \rfloor$  where  $\tau_s$  is duration of each time segment [seconds]. For our work all videos have an embedded time stamp in the bottom-right corner of the frame. We excluded the time stamp region to avoid

misclassifying changes in time as child motion. An example of our motion detection is shown in Figure 4.2.



Fig. 4.2. Example of motion detection: Preprocessed image (left), background model (middle), and moved pixels denoted in white (right).

#### 4.1.2 Reference Size Using Head Detection

The number of moved pixels  $m_j$  for time segment  $j$  is dependent on the distance between the camera and the child. A camera closer to a child will result in more moved pixels than a camera farther away because the child is contained in a region that has more pixels. To address this “scaling” issue, we scaled and limited  $m_j$  based on the size of the child. Obtaining the child body size is challenging compared to detecting the head region. The body pose can produce different shapes compared to the head and often the body is fully or partially covered with a blanket or other bed clothing. We detect the head size instead of the entire body size assuming that the two are roughly proportional. We will do this using deep learning.

Object detection performance has been significantly improved using deep learning approaches such as the Region-based Convolutional Neural Network (R-CNN) [151], the Fast R-CNN [152], and the Faster R-CNN [153]. Recent work for detecting human heads [154] is based on a R-CNN object detector [151] together with contextual information. We used the Faster R-CNN since it is one of the most effective object detectors [153]. The network is composed of three main parts: a feature extractor, a region proposal network (RPN), and a softmax classifier. The feature extractor

consists of a set of convolutional filters followed by non-linear layers that extract visual information such as color or edges. The Zeiler and Fergus (ZF) [155] network is selected as a feature extractor because it has a small number of parameters (5 convolutional layers). The RPN uses the information provided by the feature extractor to detect regions of interest where a head might be located. Then, the classifier outputs confidence values for detected regions. The confidence value ranges from 0 to 1 where a confidence of 1 represents that the network is almost certain that the region contains a head.

We trained the network using the Casablanca dataset [154] This dataset consist of 1,466 grayscale images with head annotations. Each annotation is a bounding box capturing the head location. We selected this dataset because it contains multiple heads in different poses and lighting conditions.

To find a head size for each child, first we detect heads from video frames captured every minute. Then we refine the detection results by discarding the objects that are above the upper bound limit ratio ( $l_u$ ) or below the lower bound limit ratio ( $l_l$ ) relative to the image width and height. To obtain detections when the child is sleeping, detection results with no motion ( $n_m = 0$ ) are selected. Among the refined head detections, the one with the highest confidence score is selected. We use the size of this selected head detection to obtain  $N_{max}$  per night.

$$N_{max} = c \cdot [w_h h_h / (w_i h_i)] \cdot w_p h_p \quad (4.4)$$

where  $c$  is the scale parameter,  $(w_h, h_h)$  is width and height of the head bounding box,  $(w_i, h_i)$  is width and height of the image, and  $(w_p, h_p)$  is width and height of the preprocessed image. Fig. 4.3 shows the example of head detections.

#### 4.1.3 Sleep Scoring

The Sadeh Sleep Scoring method is commonly used for scoring the Actigraphy motion index [148]. The actigraphy motion index ranges from 0 to 400. In order to





Fig. 4.3. Examples of head detections of two different infants.

have our video-based motion index  $m[j]$  be in the same range, we limit and scale each  $m_j$ .

$$m[j] = 400 \cdot (\min(m_j, N_{max}))/N_{max} \quad (4.5)$$

where  $m[j]$  is the motion index for time segment  $j$ , and  $m_j$  and  $N_{max}$  are described in Section 4.1.2. The motion index from actigraphy and video are similar measurements in the sense that more motion produces a higher motion index value. The motion index obtained from the actigraphy is based on a “zero-crossing method” which counts the number of times per each time interval that the activity signal level crosses zero (or very near zero) [149]. This indicates the amount of motion as how frequent the activity is within each time interval. The video-based motion index is obtained from the number of moved pixels as in (4.5). Due to this difference, we need to limit and scale the data to use the Sadeh’s method to the motion index obtained from auto-VSG .

We then label each time segment as sleep/awake by using the Sadeh Sleep Scoring method to  $m[j]$ . We defined the sleep onset time as the start of sleep duration which is the first consecutive sleep segments longer or equal to 5 minutes. We defined the sleep offset time as the end of sleep duration which is the last consecutive sleep segments longer or equal to 5 minutes. Duration of sleep is the time duration [minutes] between sleep onset and sleep offset. Duration of awake is the awake time [minutes] within the

duration of sleep. Since Sadeh’s method uses 11-minute window for each data point, we did not use the first and the last 5-minute data of each night for obtaining the sleep onset/offset.

Table 4.1.  
Auto-VSG ( $c = 5$ ) vs. B-VSG Labeling.

|                        | Sleep onset time     | Sleep offset time    | Awake duration         | Sleep duration         |
|------------------------|----------------------|----------------------|------------------------|------------------------|
| Sleep estimate         | Mean (SD)<br>[HH:MM] | Mean (SD)<br>[HH:MM] | Mean (SD)<br>[minutes] | Mean (SD)<br>[minutes] |
| B-VSG Labeling         | 21:01 (1:16)         | 7:32 (0:55)          | 19.07 (24.23)          | 617.86 (54.07)         |
| Auto-VSG               | 20:54 (1:11)         | 7:32 (0:51)          | 18.14 (16.77)          | 624.43 (51.02)         |
| Paired $t$ test        | $t(13)=1.01$         | $t(14)=0.01$         | $t(13)=0.14$           | $t(13)=-0.59$          |
| $\uparrow$ TOST(+30)   | $t(13)=-1.92$        | $t(14)=-2.28^*$      | $t(13)=-6.90^{**}$     | $t(13)=-1.72$          |
| $\downarrow$ TOST(-30) | $t(13)=1.23$         | $t(14)=2.27^*$       | $t(13)=6.49^{**}$      | $t(13)=2.69^*$         |

Table 4.2.  
Auto-VSG ( $c = 1$ ) vs. Actigraphy.

|                        | Sleep onset time     | Sleep offset time    | Awake duration         | Sleep duration         |
|------------------------|----------------------|----------------------|------------------------|------------------------|
| Sleep estimate         | Mean (SD)<br>[HH:MM] | Mean (SD)<br>[HH:MM] | Mean (SD)<br>[minutes] | Mean (SD)<br>[minutes] |
| Actigraphy             | 21:03 (1:02)         | 7:06 (0:54)          | 128.40 (54.27)         | 474.13 (47.2)          |
| Auto-VSG               | 20:57 (1:10)         | 7:20 (0:46)          | 140.00 (89.46)         | 482.27 (101.12)        |
| Paired $t$ test        | $t(14)=1.27$         | $t(14)=-1.58$        | $t(14)=-0.60$          | $t(14)=-0.32$          |
| $\uparrow$ TOST(+30)   | $t(14)=-1.99$        | $t(14)=-1.37$        | $t(14)=-0.80$          | $t(14)=-0.83$          |
| $\downarrow$ TOST(-30) | $t(14)=1.34$         | $t(14)=3.69^{**}$    | $t(14)=1.80$           | $t(14)=1.46$           |

## 4.2 Experimental Results

Our sleep dataset consists of 30 different nights from 30 participants. The videos recordings are for children from 9 to 30 months. The sleep data that we used is approved by the Purdue University Institutional Review Board. The data includes RGB / Infrared videos with spatial resolutions of  $320 \times 240$  pixels at 13-16 fps or  $640 \times 480$  pixels at 7-10 fps. The entire night is recorded as a sequence of videos with time stamps embedded in the video and the length of each video is 10 minutes and 14 seconds. The motion index recorded by the ankle actigraphy and B-VSG labels which includes Sleep Onset/Offset Time are also provided. The recording duration for three different methods differ—usually the video data was available only during the bed time and the actigraphy data was available for both day and night. We only used the recordings with all three methods available.

The parameters of our method (Section 4.1.1) are chosen empirically:  $w_p = 160$ ,  $h_p = 120$  [pixels],  $\tau_h = 5$  [seconds],  $T = 30$  [levels] (11.76 % of the color intensity) and  $\tau_s = 60$  [seconds]. For head detection (Section 4.1.2), we used  $l_l = 0.1$ ,  $l_u = 0.3$  and  $c = 5$  or  $c = 1$ . We did not have head annotations for our test videos so we checked the head detection result for each night through visual inspections to select the ones with correct bounding box. Among 30 nights, the head detection and refinement gave no detection for 5 nights, detected completely wrong object for 6 nights, detected near the head but with wrong size or slightly off the location for 4 nights, and gave good detection result for the rest of 15 nights. The main reason for poor head detection performance is due to the gap between the training videos and the test videos. Among the publicly available dataset that includes head annotations, we have chosen the one that is close to ours but still there were big differences—adult videos with day time scenes versus sleeping kids. For future work, having training dataset that better match the test set will improve the head detection performance. Since our focus of study is sleep detection but not the head detection itself, we used result for those 15 nights with correct head detection for our statistical analysis. The average head

region ratio  $w_h h_h / (w_i h_i)$  in (4.4) for 15 heads was 0.03 with standard deviation 0.01. It ranged from 0.01 to 0.06.

Our “B-VSG labeling” includes sleep onset/offset time and awakenings. To assess similarities among B-VSG labeling, actigraphy, and auto-VSG methods, we did paired  $t$  tests and the two one-sided  $t$  tests (TOST) for four sleep estimates: sleep onset time, sleep offset time, awake duration, and sleep duration [156]. In Table 5.2 and 5.3,  $\uparrow$  is the upper bound,  $\downarrow$  is the lower bound, \* indicates  $p < 0.05$  and \*\* indicates  $p < 0.01$ . Upper and lower TOST were completed only if the paired-sample  $t$  test did not indicate a significant difference between the measurement methods. When upper- and lower-bound TOST are significant, it demonstrates that 90% of the difference between the sleep recording methods are within the specified range of  $\pm 30$  minutes, thus implying equivalence [156]. As shown in Table 5.2, auto-VSG and B-VSG Labeling estimates have comparable agreement for all four estimates. The TOST approach indicated that the sleep onset and sleep duration estimates were not equivalent. One large outlier in sleep onset appeared to have atypical sleep architecture—a long delayed sleep. Another large outlier both in sleep offset and sleep duration was the case the baby barely moves in awake states with eyes open. Actigraphy data is sensitive to small motions during the sleep that it tends to detect more awakenings compared to B-VSG Labeling. By adjusting  $c$  to smaller value—making auto-VSG method more sensitive to motions, auto-VSG and Actigraphy estimates have comparable agreement for all estimates and is shown in Table 5.3. The TOST approach indicated that none of the estimates are equivalent in this case.

### 4.3 Conclusions

Auto-VSG has the potential to serve physicians, interventionist, and researchers in the sleep field. Auto-VSG is a minimally evasive tool that can provide sleep and awake estimates comparable to those of B-VSG. Comparisons with actigraphy were not as promising and head detection only succeeded for 50% of nights; therefore, fur-

ther auto-VSG system development is recommended before direct clinical application. However, the present study provides preliminary evidence for the use of auto-VSG in a home setting.

## 5. CLASSIFICATION OF SLEEP VIDEOS USING DEEP LEARNING

### 5.1 Introduction

Videosomnography (VSG) is a sleep analysis method which includes the labeling of sleep vs. awake intervals from video [2, 3, 157, 158]. VSG is commonly used for infants/toddlers or children with sensory sensitivities because their compliance rates with other (more invasive) sleep analysis methods can be low [2, 157, 158]. Traditional behavioral videosomnography (B-VSG) includes manual labeling of video segments as “sleep” or “awake” by a trained technician [3]. B-VSG is not used to label sleep stages (e.g., slow wave or REM sleep), rather it solely labels whether a subject is asleep or awake during a particular segment. B-VSG labeling is time consuming and expensive, because of this it has had limited use within the pediatric sleep medicine field. Polysomnography (PSG), which monitors many body functions including brain (EEG), eye movements (EOG), and heart rhythm (ECG) is the gold standard for sleep analysis but it does not capture typical sleep well [157, 159]. It is expensive and pediatric use can have low compliance. For this reason, PSG is not the most common sleep method used in homes or research.

In this chapter we describe an automated VSG method, also known as auto-VSG, to replace or assist B-VSG, while maintaining high levels of accuracy. It is important to note that our goal is to label each frame of a sleep video with the label “sleep” or “awake.” In this work we are not interested in labeling sleep stages, such as REM sleep.

Auto-VSG is a growing area in sleep analysis with preliminary studies using signal/image processing systems that use motion during sleep for sleep/awake labeling [2, 93–95, 160]. In these studies, motion is estimated using frame differenc-

ing [93,94] or motion vectors [2,95,160]. However, each of these studies were conducted in a controlled setting and do not account for the wide range of camera positions and lighting variations that are common among in-home VSG recordings. In our work, we use deep learning approaches to classify in home sleep videos as sleep vs. awake that adjust for these ‘in the wild’ factors.

In this chapter, we propose a new approach for sleep video analysis. The contributions in this chapter are: (1) we describe the key factors in sleep video classification (i.e., movements over long period of time) that are not addressed in commonly used action classification problems (Section 5.2) (2) we propose a sleep/awake classification system with a recurrent neural network using simple motion features (Section 5.3) (3) we experimentally show our system successfully learns long-term dependencies in sleep videos and outperform one of the recent method that has been successful in public action dataset (Section 5.4).

## 5.2 Related Work

### 5.2.1 Motion and Long Term Dependencies in VSG

We assume that there is less motion of the subject during sleep than when awake [149]. One simple way to classify sleep vs. awake is to set a static threshold based on the assumption that more motion in a frame is awake and less motion is sleep. However, sleep and awake patterns are not that simple.

Typically in VSG, sleep onset is established based on information from more than 20 minutes of observed video and awakenings must include purposeful movements and be more than one minute in duration. Similarly, actigraphy methods use both a motion index (the amount of motion within a time segment [149]) and information about the duration before and after the target minute [148]. Both movement and temporal information are needed to accurately capture sleep and awake states.

### 5.2.2 Long Short- Term Memory Networks (LSTM)

A Recurrent Neural Network (RNN) is a deep learning network used for processing sequential data by forming a memory through recurrent connections from the previous inputs to the current output [161, 162]. Similar to Convolutional Neural Network (CNN) spatially sharing parameters, a RNN temporally shares parameters assuming that the same parameter can be used for different time increments (i.e., the conditional probability distribution over the variables at time  $t+1$  given the variables at time  $t$  is stationary) [163].

For a standard RNN, the range of input sequence that can be accessed is quite limited in practice because of the “vanishing gradient” effect (VGE) [162, 164]. VGE is a problem that gradients propagated over many recurrent connections tend to vanish mainly due to the exponentially smaller weights given to long-term interactions (involving multiplication of many Jacobians) compared to short-term ones [163]. Long Short-Term Memory Networks (LSTM) [164, 165] is a special type of RNN which enables long-range learning by reducing VGE. LSTM uses a structure known as gates that can regulate the removal or addition of information. It is based on the idea of creating paths through time that have derivatives that neither vanish nor explode [163]. While the repeating module in a standard RNN contains a single layer, the one in LSTM contains four interacting layers—forget gate layer, input gate layer, update layer, and output layer. LSTM has been widely used for processing various sequential data and has been successful in language processing such as speech recognition, text recognition, and machine translation.

### 5.2.3 Video Classification Using Deep Learning

Image classification using deep learning began in 2012 with the ImageNet challenge [166], video classification using deep learning is still in the early stages with several recent studies focused on specific public datasets [167–170]. These methods make use of the basic idea in Convolutional Neural Networks (CNN) classification



approaches in still images to solve video classification problems. Karpathy *et al.* [167] presented slow fusion methods for large scale video classification using CNNs. This was one of the early works on deep learning video classification to extend the connectivity of the CNN in the time dimension to learn spatio-temporal features. Another approach incorporating temporal information is the Long-term Recurrent Convolutional Networks (LRCN) proposed by Donahue *et al.* [168, 169]. They first obtained visual features from each frame using a conventional CNN and then used the features as inputs to the recurrent models (see Figure 5.1). As shown in Figure 5.1,

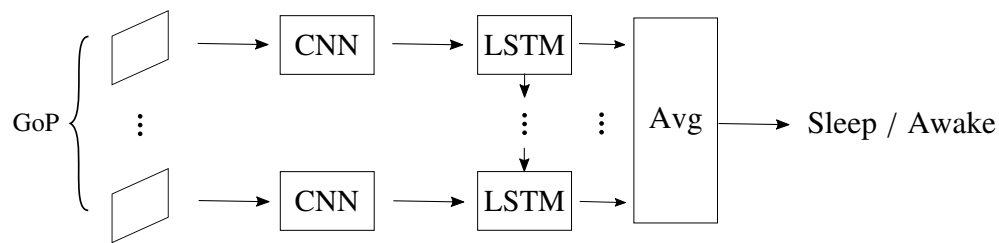


Fig. 5.1. LRCN [168, 169]. GoP is a Group of Pictures.

LRCN [168, 169] combines visual features (output from each CNN) with sequence learning (LSTM). The advantage of having this structure is that it can learn unique appearance in video while also learning temporal patterns of variable lengths. In LRCN, the spatial and temporal information is processed in two separate steps—first each frame (spatial information) goes into CNN and outputs feature vectors, then series of feature vectors (temporal information) go through LSTM. Due to these separated steps, there is a limit to learning spatial changes over time (i.e. motion). Another approach for human action recognition is spatio-temporal CNN filters (C3D) proposed by Tran *et al.* [170]. C3D extends the conventional CNN with an additional temporal dimension by using 3-dimensional CNN kernels in all convolutional layers (see Figure 5.2). C3D [170] in Figure 5.2 uses one network to learn both spatial and temporal information at the same time by using 3-dimensional convolution kernels that include the temporal dimension. This network can learn motion changes over time, but with limited temporal range (e.g. the length is fixed to 16 consecutive

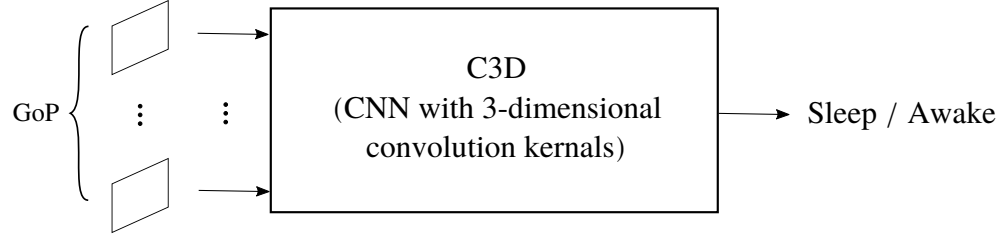


Fig. 5.2. C3D [170]. The C3D convolution kernel includes temporal depth in addition to 2-dimensional CNN kernel of width and height.

frames at a time). It was reported in [170] that C3D performed similar or better compared to other methods including deep networks [167] and LRCN [168] on an action recognition public dataset UCF101 [171]. While there has been improvements for specific action recognition datasets, whether these methods can be generalized for use in other types of video classification problems is an open question. Public action recognition datasets used in all of the above mentioned studies were short video sequences with repetitive and unique action in each class (e.g. the action classes in UCF101 dataset include *apply eye makeup*, *baby crawling*, *brushing teeth*, *horse race*, *knitting*, etc.). Sleep videos are much longer in length (up to 8-9 hours) and tend to have relatively few actions. Also, the appearances (e.g. human or objects appearing in the scene) change only slightly between the sleep/awake states.

### 5.3 Proposed Method

In this section we describe our proposed method for labeling frames of a sleep video as “sleep” or “awake” from RGB/infrared videos using motion information. Figure 5.3 shows our system. First, we define consecutive video frames in small groups as Group of Pictures (GoP). The proposed system uses frame differencing within GoP to obtain motion information (described in detail in Section 5.3.1) and two-layer LSTM architecture to incorporate information from previous video GoPs.

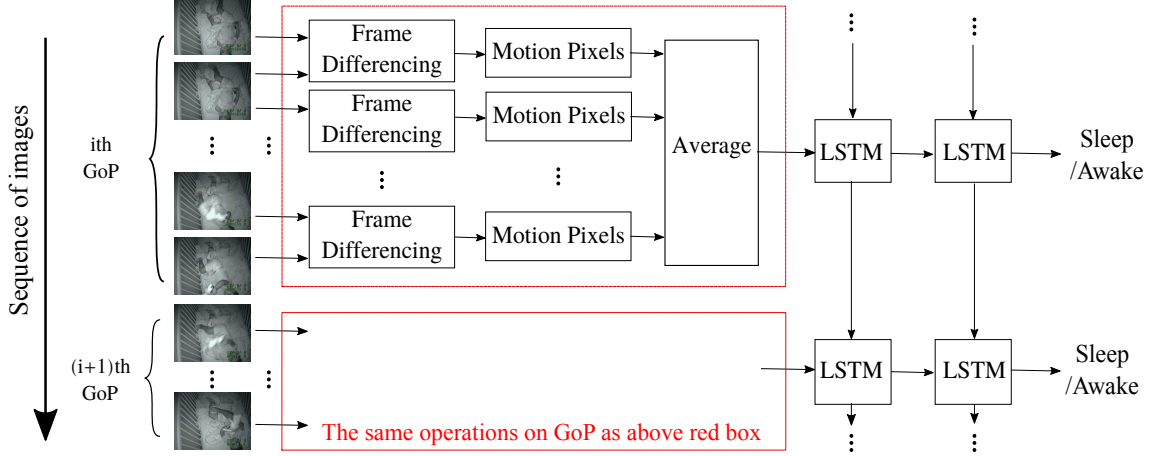


Fig. 5.3. Proposed Sleep Detection System: Sleep/Awake Using a Motion Index and LSTM.

### 5.3.1 Motion Detection/Motion Index

We shall assume that the child is the only source of motion in the sleep video. Background subtraction is widely used for detecting movements from static cameras [150]. One of the simple background subtraction methods is frame differencing which detects motion in frame by taking the difference (subtraction) between the current frame and the previous frame (the background model). For a sequence of gray scale images in GoP at constant frame rate and size, we take the frame difference in each consecutive pair. This difference indicates whether each pixel in the frame is classified as “moved” or “not moved.” A pixel is classified as “moved” if Equation (5.1) is true

$$|I_{i-1}[x, y] - I_i[x, y]| > T \quad (5.1)$$

where  $I_i[x, y]$  is a pixel in frame  $i$ , and  $T$  is a threshold for determining movement for one pixel. For our experiments the value of  $T$  is empirically determined and the value we use is described in Section 5.4.2. We quantify the amount of motion as the number of pixels classified as “moved.” We define the motion index for a GoP as the average the amount of motion for each frame pair in the GoP. The red box shown in

Figure 5.3 is the motion detection block and the output of this block is the motion index for each GoP.

We minimize the use of the empirically driven parameters (only using one parameter  $T$ ) by using deep learning methods that learn the sleep vs. awake patterns based on the motion index.

### 5.3.2 Loss Function

For an imbalanced dataset where one class has much larger number of samples than the other class, the trained model can be biased toward the class in the majority. A typical sleep video dataset is imbalanced where the number of sleep labels dominates awake labels. To compensate for this data imbalance, class-wise weights can be set in the loss function. We define the weight  $w_j$  for the class (sleep or awake)  $j$  as

$$w_j = 1 - \frac{n_j}{\sum_j n_j} \quad (5.2)$$

where  $n_j$  is the number of samples in class  $j$ . The idea is that when there is more data for class  $j$ , a smaller weight is assigned. Using these weights, the weighted softmax cross entropy loss function for sequence data  $(x_0, y_0), \dots, (x_i, y_i), \dots, (x_n, y_n)$  is defined as

$$L = \sum_i w_{y_i} \left( -\log \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \right) \quad (5.3)$$

where  $y_i$  is the actual class index for the sample  $x_i$ , and  $f_j$  is the predicted probability of  $x_i$  belonging to class  $j$ . With uniform weights across classes ( $w_0 = \dots = w_j = \dots$ ), the loss function  $L$  becomes the regular softmax cross entropy.

## 5.4 Experiments

### 5.4.1 Dataset

Our sleep dataset consists of in home sleep videos of 30 different nights from 30 children. The sleep videos are for children from 9 to 30 months of age. Each night

is a different sleep video sequence of a different child. Our total data set consists of 30 children for 30 nights. The camera used for recording is Swann ADW-400 Digital Guardian Camera & Recorder. It records in color mode during the day and switches to black and white/infrared mode at night. This project was approved by the Purdue University Institutional Review Board (IRB). The sleep videos have spatial resolutions of  $320 \times 240$  pixels at 13-16 frames/s (fps) or  $640 \times 480$  pixels at 7-10 fps. The entire night is recorded as a sequence of videos with time stamps embedded in the video frame and the length of each video is 10 minutes and 14 seconds. Along with the sleep videos, B-VSG labels for sleep onset, offset, and awakenings were used in the analyses. This information was obtained as ground truth from trained observers. We did not use the audio due to too much noise in the signal.

For preprocessing, videos were sub-sampled at 4 fps. Then, the GoP (16 frames) were obtained. While the B-VSG labels are in units of minutes, a GoP in our settings corresponds to a 4-second duration. GoPs that do not fully belong to sleep or awake (i.e., partly Sleep and partly Awake GoPs), were not used in the experiment. How we divided the sleep dataset into training and testing sets is shown in Table 5.1. As

Table 5.1.  
Training/Test Set Division of Sleep Dataset.

| Sleep Dataset       | # GoPs<br>for Sleep | # GoPs<br>for Awake | # GoPs<br>in total |
|---------------------|---------------------|---------------------|--------------------|
| Train (20 children) | 179,108             | 13,691              | 192,799            |
| Test (10 children)  | 88,234              | 7,781               | 96,015             |
| Total (30 children) | 267,342             | 21,472              | 288,814            |

we can see from Table 5.1, there is an imbalance between the two classes of “Sleep” and “Awake”. For the training set of 20 children, the number of continuous sequences were 33 and the length ranged from 378 to 11,352 GoPs. In case where a child had

some “out of bed” time, the corresponding GoPs were excluded from our training/test sets hence resulting in multiple sequences for one child.

#### 5.4.2 Implementation Details

For the motion index threshold  $T$  we used  $T = 30$  (11.7% difference in gray scale intensity levels) and image size of  $320 \times 240$  in gray scale for obtaining the motion index for all the GoPs. The value of parameter  $T$  was empirically determined. For training on very different dataset, such as videos with lower contrast,  $T$  can be set to lower values. Our Long Short-Term Memory Network (LSTM) described in Section 5.2.2 was implemented using Python and TensorFlow. For the LSTM, we used a hidden unit size of 128 and 2 layers of cells with dropout layers with probability of 0.5. The softmax cross entropy loss was used as the cost function for training. The Adagrad (Adaptive Gradient) method [172] was used for gradient descent optimization. To reduce computational complexity, we organized all the training set as a sequence of GoPs and put them in mini-batches of size 30. Since the number of training GoPs is 192,799 and is not dividable by our batch size 30, the remaining last 19 GoPs were discarded hence resulting in a  $30 \times 6,426$  matrix. The GoPs in the first column (the first batch) were assumed to be the start of each sequence although it would not exactly match with the actual start of the sequence for each child. The size of each mini-batch was  $30 \times (\text{back propagation window size})$ . The network is initialized with a vector of zeros and gets updated after reading each GoP. The number of epochs we used for training is 10.

For training and testing the spatio-temporal CNN filters (C3D), we used the implementation provided in Caffe [170].

#### 5.4.3 Results

To assess the performance, we used five metrics, which are Accuracy,  $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ , Precision,  $PRE = \frac{TP}{TP+FP}$ , Recall,  $REC = \frac{TP}{TP+FN}$ , Specificity,

$SPEC = \frac{TN}{TN+FP}$  and Cohen’s kappa ( $\kappa$ ). The test dataset is shown in Table 5.1. Table 5.2 and Table 5.3 show the results for different classification models of on the same test set. Table 5.2 is the result using uniform weights for the loss function. In this table, C3D-f is C3D pre-trained on Sports-1M dataset [167] and finetuned on our sleep dataset. C3D-t is C3D trained on our sleep dataset from scratch. Our proposed models are denoted as LSTM- $k$ , where the number  $k$  refers to the size of the back propagation window in the unit of number of GoPs that is used during training the model. The duration of one GoP in unit of seconds is  $16 \text{ frames}/4 \text{ fps} = 4 \text{ seconds}$  (i.e., 5 GoPs are 20 seconds, and 30 GoPs are 2 minutes). Table 5.2 shows the result

Table 5.2.

Results [%] for the number of test GoPs  $n = 96,015$ . Models trained using loss with uniform weights.

| Model         | $ACC$        | $PRE$        | $REC$        | $SPEC$       | $\kappa$ |
|---------------|--------------|--------------|--------------|--------------|----------|
| C3D-f         | 84.25        | 93.52        | 89.03        | 30.07        | 0.15     |
| C3D-t         | 89.54        | 93.29        | 95.49        | 22.05        | 0.20     |
| <b>LSTM-5</b> | <b>95.60</b> | <b>96.43</b> | <b>98.87</b> | <b>58.55</b> | 0.66     |
| LSTM-15       | 92.89        | 92.87        | 99.93        | 12.98        | 0.21     |
| LSTM-30       | 94.31        | 94.83        | 99.23        | 38.62        | 0.50     |
| LSTM-50       | 92.77        | 92.71        | 99.99        | 10.90        | 0.18     |
| LSTM-75       | 93.51        | 93.61        | 99.74        | 22.85        | 0.34     |
| LSTM-85       | 92.53        | 93.56        | 98.66        | 22.95        | 0.30     |

for models trained with regular softmax cross entropy loss function. Compared to using one GoP at a time for classification (i.e., C3D-f and C3D-t), using multiple GoPs (i.e., LSTM- $k$ ) improved accuracy while maintaining high recall. LSTM-5 improved the performance across all four metrics. However, the specificity is low due to the data imbalance. Since the model is trained to minimize the overall loss including both sleep and awake GoPs, the specificity that involves only awake GoPs is not giving

consistent results. Next, Table 5.3 shows the result for models trained using weighted loss as described in Section 5.3.2. We can see that the specificities are improved.

Table 5.3.  
Results [%] for the number of test GoPs  $n = 96,015$ . Models trained using weighted loss.

| Model   | <i>ACC</i> | <i>PRE</i> | <i>REC</i> | <i>SPEC</i>  | $\kappa$ |
|---------|------------|------------|------------|--------------|----------|
| LSTM-5  | 93.33      | 97.81      | 94.87      | <b>75.88</b> | 0.61     |
| LSTM-15 | 93.46      | 97.75      | 95.06      | <b>75.22</b> | 0.62     |
| LSTM-30 | 95.47      | 96.70      | 98.44      | <b>61.88</b> | 0.67     |
| LSTM-50 | 95.58      | 95.79      | 99.57      | 50.31        | 0.63     |
| LSTM-75 | 20.48      | 98.62      | 13.66      | 97.83        | 0.02     |
| LSTM-85 | 92.69      | 93.85      | 98.51      | 26.74        | 0.34     |

There are good agreements between traditional B-VSG and our proposed methods (LSTM-5 on loss with uniform weights,  $\kappa = 0.66$ ; LSTM-5/15/30/50 on weighted loss,  $\kappa > 0.6$ ) and fair to poor agreements ( $\kappa < 0.4$ ) on the rest of the methods including C3D.

Figure 5.4 and 5.5 are the ROCs [173]. Unlike accuracy and precision, ROCs are insensitive to changes in class distribution since it is based upon True Positive rate and False Positive rate. Note that in Table 5.2 and Table 5.3 recall and specificity are obtained based on the discrete outputs generated with a threshold of 0.5—we take the predicted class as the one with the higher probability. For the ROCs, we used the monotonicity of thresholded classifications [173]. Figure 5.4 shows that all the proposed methods (LSTM- $k$ ) have higher Area Under the ROC (AUC) than the C3D models. Except for the AUC drop at  $k = 85$  due to the long back propagation stages in the training, the rest of all the LSTM- $k$  models have AUC higher than 0.85. C3D-t and C3D-f have AUC of 0.62 and 0.65 respectively both giving much lower performance compared to the proposed methods. LSTM-75 where 75 GoPs corresponds to



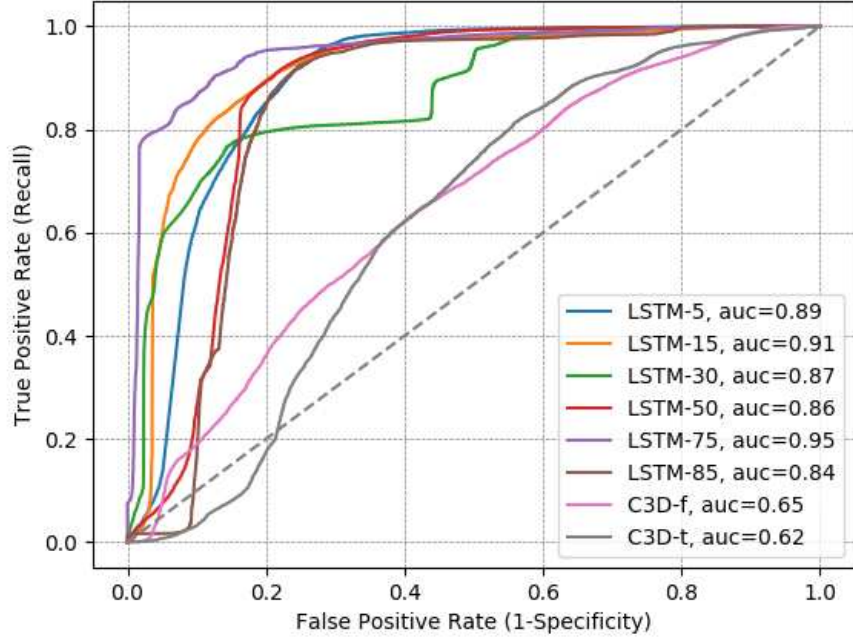


Fig. 5.4. ROCs. Models trained using loss with uniform weights. GoP is Group of Pictures.

5-minute duration gave the highest performance (AUC=0.95). Figure 5.5 shows the result for the models trained using the weighted loss described in section 5.3.2. The classifier fails for  $k=85$  more severely compared to the uniform weight cases of the same  $k$  value. For the models trained using the weighted loss, LSTM-50 (duration of 3 minutes and 20 seconds) gave the highest performance (AUC=0.95). The overall results show how much the long-term temporal motion information plays a significant role in sleep vs. awake classification. This is surprising given the fact that the proposed method used minimal visual information of only one motion index for each GoP. The proposed methods outperforms the general video classification method by modeling the long-term motion patterns in sleep videos.

As described in Section 5.2, C3D is good for classifying unique appearance and short action in each class by learning spatio-temporal features in videos but due to

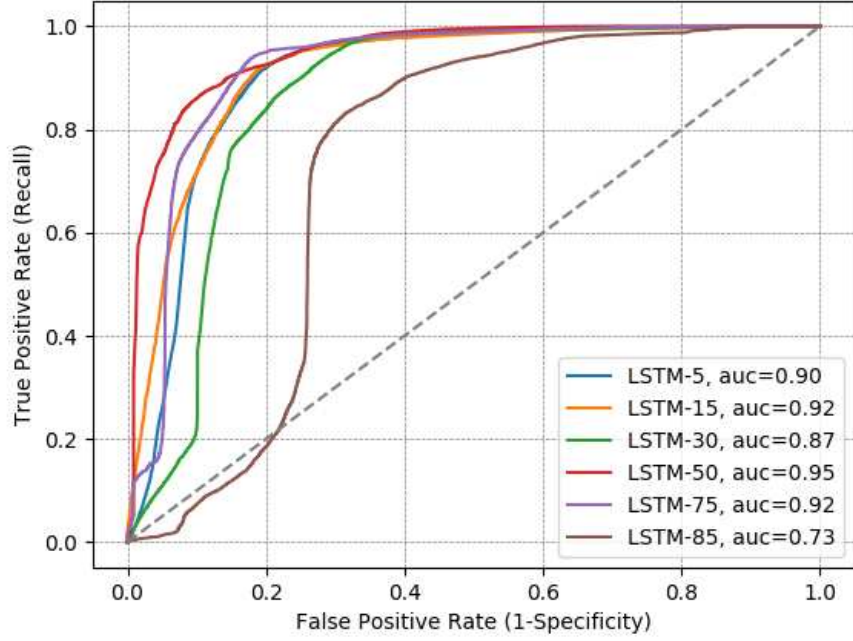


Fig. 5.5. ROC curve. Models trained using weighted loss. GoP is Group of Pictures.

the limited temporal range it takes, C3D did not work well for classifying sleep videos that have long temporal dependencies. Also, due to the slight changes in appearance between the sleep/awake states and few actions in sleep videos, learning appearance pattern in C3D did not contribute well on improving the overall performance. Our proposed method enabled capturing the temporal history of motion changes by using LSTM on sequence of GoPs and simple motion feature for each GoP.

## 5.5 Conclusions

In this chapter, we described a system for sleep vs. wake classification based on our observation that long temporal information is important. From the prior knowledge that motion is the key factor for determining sleep versus awake in B-VSG, we described a motion index to summarize the motion information for each

GoP and then combined this with the recurrent model to label each GoP as asleep or wake. Our experiment demonstrated interesting results that using LSTM with simple motion feature for GoP outperformed one of the latest general video classification methods for sleep vs. awake video classification. We also showed how weighting the loss function can affect various performance metrics for imbalanced sleep dataset (i.e. the increase in specificity).

The design of our system is based on the prior knowledge in sleep medicine (i.e. the motion changes over long duration is the key factor in determining sleep vs. wake) and in signal processing (i.e. methods for simple motion feature in video). For future work, more general video classification methods that require less prior knowledge should be further investigated.

## 6. IMAGE-BASED GEOGRAPHICAL LOCATION ESTIMATION USING WEB CAMERAS

### 6.1 Sunrise/Sunset Estimation

Sunrise and sunset can be obtained by classifying each image from the camera with the label “Day” or “Night.”

One of the factors that can be used for detecting Day/Night is the brightness of the image. In [116], the mean of the combined RGB components were used to detect Day. In our work, we used the luminance to estimate the brightness of the image. We first convert from the *RGB* to *YCbCr* color space and use the *Y* component to obtain the average luminance. We assume that an image with large luminance tends to be Day. We have ignored camera AGC effects. We recognize that this introduces error in our estimates for sunrise and sunset due the fact that the images will be “brighter” than normal. In our operational scenario we have no control of this in that we cannot turn off the camera AGC.

The color of the sky is mostly sensitive to whether it is Day or Night while other objects in the scene can have various colors. To make use of this fact in determining Day/Night, we can define into two spatial regions—the sky region and the non-sky region. We are assuming that some part of the field of view of the camera “sees” the sky. The set of pixels in an image belonging to the sky is defined as the sky region, and the rest of the pixels which do not belong to the sky are defined as the non-sky region. Day/Night detection based on the luminance of the entire image could be incorrect due to factors in the non-sky region, e.g. lights from a building at night, the dark objects or shadows that appears during the day. Therefore, it is more accurate to focus on the sky region for Day/Night detection. In [116] the entire RGB image

was used and in [118] the sky regions were detected using a camera with a field of view from the dash of a vehicle using the road information.

Our method is a variation of the above in that we focus on the sky region and find the mean of  $Y$  in the sky region:

$$Y_{sky.i} = \frac{1}{M} \sum_{j=0}^{M-1} Y_{sky.i,j} \quad (6.1)$$

where  $Y_{sky.i}$  is the mean sky luminance of the  $i_{th}$  image and  $Y_{sky.i,j}$  is the luminance of the  $j_{th}$  pixel in the sky region of the  $i_{th}$  image.  $M$  is the number of pixels in the sky region. Here we assume that the camera is static and the sky region for the camera remains the same for all the images. Our approach to sky detection is discussed in Section 6.2.

We will estimate sunrise and sunset by detecting transitions from Night to Day and Day to Night. To detect Day/Night transitions from the luminance of the sky region, a threshold must be determined. If we assume the images are obtained over a 24-hour period, we know that approximately a quarter of the images are either Day images or Night images if the camera is located in the latitude range between  $60^\circ S$  and  $60^\circ N$ . Since  $Y_{sky.i}$  has large value for Day and small value for Night, we can find a threshold for  $Y_{sky.i}$  to label the image as Day or Night. Two different thresholds for classifying Day/Night can be used:

$$th_{mean} = \frac{1}{N} \sum_{i=0}^{N-1} Y_{sky.i}, \quad (6.2)$$

$$th_{mid} = \frac{\max\{Y_{sky.i}\} + \min\{Y_{sky.i}\}}{2}. \quad (6.3)$$

where  $N$  is the number of images. In [116] the  $th_{mid}$  was used for the threshold but when we used it to our experimental work,  $th_{mean}$  provided better results. If the mean luminance of the sky region of an image is larger than the threshold, we classify it as Day, otherwise we classify it as Night.

From the sequence of images denoted as either Day or Night, we can denote the times where the transitions between Day and Night labeled images occur. If the

labels change from Night to Day, we can estimate that the sunrise occurs within the time interval between those consecutive images, if the labels change from Day to Night, we can estimate that the sunset occurs within the time interval between those consecutive images.

We then estimate the sunrise as the time of the start of Day. In this case, the accuracy of the sunrise estimation depends on the sampling interval of images. If the images are sampled every  $s$  minutes, the error of sunrise would be less than  $s$  minutes. Likewise, we can approximate the sunset as the time of the start of Night. The error of sunset would also be within  $s$  minutes. If the estimated Day/Night labels are accurate, exactly one start of Day and one start of Night should occur during the image sequence of 24-hour period. Due to the dynamic weather conditions, some images can be falsely labeled as Night during the day. One way to eliminate these outlier images would be taking the earliest start of Day as sunrise and taking the latest start of Night as sunset.

## 6.2 Sky Region Detection

There are many methods for detecting the sky region in an image. In [118] sky detection is only considered for the special case where the images are the front view from dash cameras in vehicles. In [117] edge detection of sky region is used to predict the solar exposure. They describe a general approach to separate the sky from the rest of the image by determining the edge of the sky region. The accumulative frame difference between an image and the successive image is used to obtain the sky region in [119]. The sky is assumed to be at the top of image and the clouds are dynamic. Using this method requires several sample images to detect the sky region. Also, it is valid only when the sample images are Day images since the method is based on the fact that the sky is dynamic compared to the foreground objects.

We propose a different approach to detect the sky region by using one image of a clear sky. By clear sky we mean no clouds in the sky and in our initial experiments this image was manually chosen. The sky detection approach we used is then:

1. Obtain an image from the blue channel of the camera.
2. Use the Canny edge operator to find edges. This will create a binary image or edge mask where edge pixels are set to 1.
3. Use morphological filtering (dilation) to close gaps in the boundaries of the edge mask.
4. Invert the dilated binary image (edge mask) where the boundary pixels are inverted from 1 to 0 and the surface pixels are inverted from 0 to 1.
5. Find the largest connected region at the top of the binary image:
  - (a) Find all the connected components in the binary image.
  - (b) Sort the connected components with respect to the number of pixels contained in descending order.
  - (c) For each of the connected components check the location of each connected component to determine whether it is at the top part of the image. If the connected component is at the top part of the image, select it as the sky region and if not, go to the next largest connected component. Repeat until the sky region is found.

The results of using the the above sky detection technique are shown in Figure 6.1.

### 6.3 Estimating Location from Sunrise/Sunset

Once the sunrise/sunset is estimated as described above we can use it to determine the camera location. In [110] they proposed what they called the CBM model to estimate the length of the day for a flat surface for a given latitude and day of the year.



Fig. 6.1. A collection of pairs of test images and their skymask.

They also described a daylength model to allow for various conditions of daylength and twilight for a full range of latitudes. Using the CBM daylength model [110] we estimate latitude by:

$$\theta = 0.2163108 + 2 \tan^{-1} [0.9671396 \tan[0.00860 \times (J - 186)]] , \quad (6.4)$$

$$\phi = \sin^{-1} [0.39795 \cos \theta] , \quad (6.5)$$

$$D = 24 - \frac{24}{\pi} \cos^{-1} \left[ \frac{\sin \frac{p\pi}{180} + \sin \frac{L\pi}{180} \sin \phi}{\cos \frac{L\pi}{180} \cos \phi} \right] , \quad (6.6)$$

where  $\theta$  is the revolution angle,  $J$  is the day of the year,  $\phi$  is the sun's declination angle,  $D$  is the daylength, and  $L$  is the latitude. By numerically solving Eq. 6.6, we can estimate latitude ( $L$ ) from daylength ( $D$ ) and the day of the year ( $J$ ). In this paper, the daylength coefficient ( $p$ ) was set to 6.0 to correspond to the daylength definition which includes civil twilight.  $D$  is the time difference between the sunrise and sunset.

Longitude can be estimated from local noon [174]. If we know UTC (Coordinated Universal Time) when the sun is at its highest point in the sky at a location on the Earth (local noon), then we can determine the time difference between the local noon



and the noon in UTC. The time difference can be converted to longitude (l) since we know that the Earth approximately rotates 15 degrees per hour.

$$l = \begin{cases} (12 - n + u) \times 15 & u \leq 12 \\ (n + u - 12) \times 15 & u > 12 \end{cases} \quad (6.7)$$

where  $n$  is the local noon and  $u$  is the UTC offset for the local area. All the variables  $l$ ,  $n$  and  $u$  are in unit of hours. The local noon can be approximately estimated from sunrise and sunset.

$$n = \frac{t_{sunset} + t_{sunrise}}{2} \quad (6.8)$$

where  $t_{sunset}$  and  $t_{sunrise}$  are the local time of sunset and sunrise in hours. Since the earth rotation is nearly constant, we assume that at the middle of the sunrise and sunset, the sun is at its highest point in the sky.

#### 6.4 Experimental Results

We evaluated our methods using 10 static IP-connected web cameras. For each camera images were downloaded every 5 minutes and stored with a timestamp based on UTC-5. The images were obtained during 21-27 December 2013 (UTC-5).

The process begins by detecting the sky region for an image from each camera as described in the Section 6.2. The output of this process is the sky mask of each camera. Next all images are converted from the *RGB* to *YCbCr* color space and the *Y* component of each image is obtained (see Section 6.1). The sky mask is then used for determining the mean sky luminance ( $Y_{sky-i}$ ) for each image. Next images are classified as Day or Night by using the threshold. After the Day or Night images are obtained, they are used to estimate the sunrise and sunset. Finally, the latitude and longitude are obtained using the estimated sunrise and sunset (see Section 6.3).

In Figure 6.2,  $th_{mean}$  and  $th_{mid}$  described in the previous section are denoted. Figure 6.2 also shows that the luminance of the sky region separates Day/Night images while the luminance of the entire image poorly separates between Day and Night images.

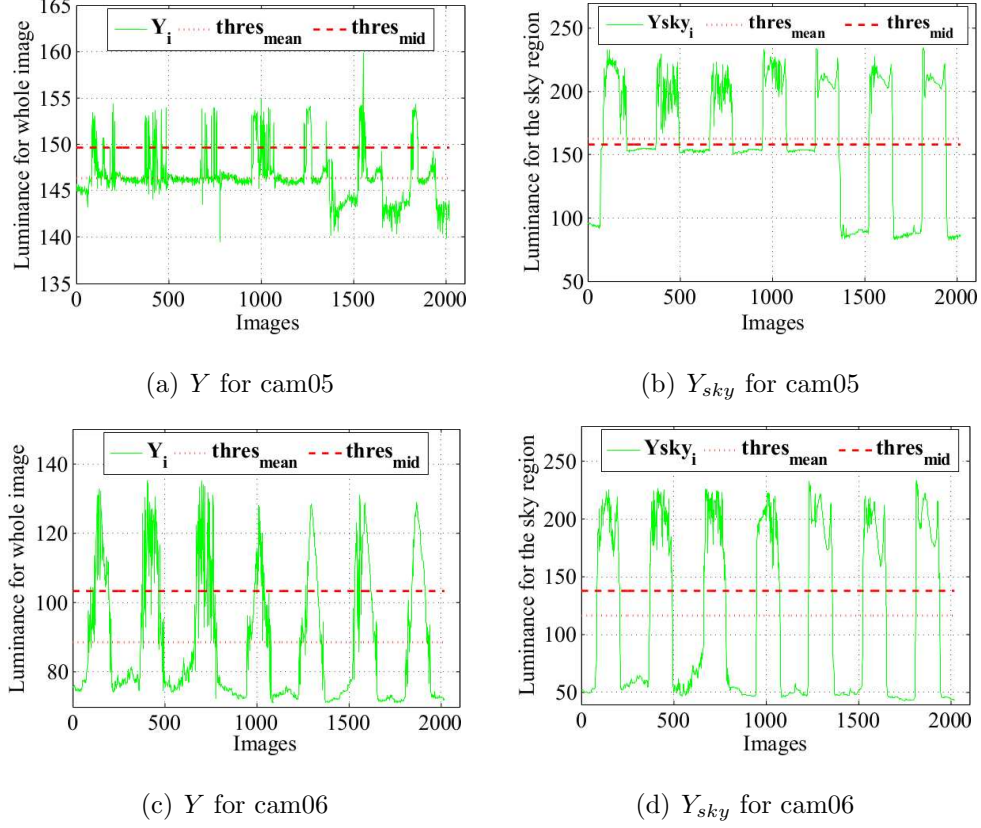


Fig. 6.2. The mean luminance of the entire image vs. the sky region.

The mean estimated sunrise/sunset is shown in Table 6.1 for camera01. We know the exact location of this camera and can find the ground truth sunrise and sunset from [175] using the latitude and longitude information of this camera (North 40 degree, 26 minutes, West 86 degree, 55 minutes). The “est. rise” and the “est. set” columns are the estimated sunrise and sunset in hh:mm. The “GT rise” and “GT set” columns are the ground truth sunrise and sunset in hh:mm rounded to the closest minute. For the ground truth, the sunrise and sunset civil twilight were used. The mean error for 7 days was -7.8 [minutes] with standard deviation of 6.7 [minutes] for the sunrise and 9.9 [minutes] with standard deviation of 5.4 [minutes] for the sunset.

In Tables 6.2 and 6.3, the “mean” and “std” columns refers to the mean and the standard deviation of latitudes for 7 days. The “GT” column refers to the ground truth. In general we do not know the exact location of some of the cameras used in

Table 6.1.  
Sunrise/sunset detection for camera01 for using  $th_{mean}$ .

| Date   | est.<br>rise | est.<br>set | GT<br>rise | GT<br>set | rise<br>error | set<br>error |
|--------|--------------|-------------|------------|-----------|---------------|--------------|
| Dec 21 | 07:55        | 17:35       | 07:37      | 17:55     | -18.4         | 20.3         |
| Dec 22 | 07:45        | 17:45       | 07:37      | 17:56     | -7.9          | 10.8         |
| Dec 23 | 07:50        | 17:50       | 07:38      | 17:56     | -12.5         | 6.4          |
| Dec 24 | 07:40        | 17:50       | 07:38      | 17:57     | -2.0          | 7.0          |
| Dec 25 | 07:50        | 17:45       | 07:38      | 17:58     | -11.6         | 12.6         |
| Dec 26 | 07:40        | 17:50       | 07:39      | 17:58     | -1.3          | 8.2          |
| Dec 27 | 07:40        | 17:55       | 07:39      | 17:59     | -0.9          | 3.9          |

Table 6.2.  
The result for latitude for using  $th_{mean}$ .

| Camera | mean[°] | std[°] | GT[°] | $e_L$ [%] |
|--------|---------|--------|-------|-----------|
| 1      | 43.6    | 2.0    | 40.4  | 1.8       |
| 2      | 32.7    | 24.3   | 41.8  | 5.0       |
| 3      | 43.7    | 3.7    | 41.8  | 1.1       |
| 4      | 41.3    | 3.5    | 40.4  | 0.5       |
| 5      | 36.3    | 3.4    | 38.0  | 0.9       |
| 6      | 36.2    | 4.9    | 38.8  | 1.4       |
| 7      | 31.5    | 2.1    | 36.1  | 2.6       |
| 8      | 32.0    | 1.5    | 36.1  | 2.3       |
| 9      | 35.1    | 25.4   | 42.4  | 4.1       |
| 10     | 26.7    | 1.2    | 34.4  | 4.3       |

our study. The “ground truth” locations we used here were obtained from their IP addresses or using Google maps. This approach is somewhat problematic but it reflects

Table 6.3.  
The result for longitude for using  $th_{mean}$ .

| Camera | mean[°] | std[°] | GT[°]  | $e_l$ [%] |
|--------|---------|--------|--------|-----------|
| 1      | -86.6   | 0.5    | -86.9  | 0.2       |
| 2      | -91.9   | 11.9   | -87.6  | 2.4       |
| 3      | -88.2   | 0.8    | -87.6  | 0.3       |
| 4      | -88.0   | 1.2    | -86.9  | 0.6       |
| 5      | -77.6   | 0.8    | -78.5  | 0.5       |
| 6      | -76.2   | 1.4    | -76.9  | 0.4       |
| 7      | -75.4   | 1.4    | -75.7  | 0.2       |
| 8      | -76.8   | 1.0    | -75.7  | 0.6       |
| 9      | -73.1   | 6.8    | -72.5  | 0.3       |
| 10     | -119.0  | 0.3    | -119.8 | 0.5       |

the nature of the problem we are trying to address. To evaluate the performance, we defined the error metrics for latitude ( $e_L$ ) and longitude ( $e_l$ ) as:

$$\begin{aligned} e_L &= \frac{|L_{est} - L_{GT}|}{180} \cdot 100 \quad [\%] \\ e_l &= \frac{|l_{est} - l_{GT}|}{360} \cdot 100 \quad [\%] \end{aligned} \quad (6.9)$$

where  $L_{est}$  and  $L_{GT}$  both in units of degree (°) are estimated and the ground truth latitudes and  $l_{est}$  and  $l_{GT}$  both in units of degree (°) are estimated and the ground truth longitudes. In these tables, we see that the amount of error  $e_L$  in latitude is larger compared with the error  $e_l$  in longitude. We discovered that for each case for Cameras 2 and 9, there is erroneous estimation of the sunrise and sunset that increases the overall error. These incorrect estimations are caused by lights in the camera field of view during the night that result in a sudden rise of luminance after the sunset hence leading to the wrong estimation of sunset.

In conclusion, we estimated the approximate location of a web cam by analyzing its images. We showed that we could effectively estimate locations with less than

2.4% error for the longitude and less than 5% error for the latitude. In future work we will investigate how we can compensate for camera AGC effects and fine grained temporal measurements.

## 7. CONCLUSION

### 7.1 Summary

In this thesis we addressed two interesting video-based health measurements. First is video-based Heart Rate (HR) estimation, known as video-based Photoplethysmography (PPG) or videoplethysmography (VHR). We adapted an existing video-based HR estimation method to produce more robust and accurate results. Specifically, we removed periodic signals from the recording environment by identifying (and removing) frequency clusters that are present the face region and background. We investigated and described the motion effects in VHR in terms of the angle change of the subjects skin surface in relation to the light source. Based on this understanding, we discussed the future work on how we can compensate for the motion artifacts. Another is Videosomnography (VSG), a range of video-based methods used to record and assess sleep vs. wake states in humans. We described automated VSG sleep detection system (auto-VSG) which employs motion analysis to determine sleep vs. wake states in young children. The analyses revealed that estimates generated from the proposed Long Short-term Memory (LSTM)-based method with long-term temporal dependency are suitable for automated sleep or awake labeling. We created web application ( Sleep Web App) that deploys our sleep/awake classifications method to serve easy accesses to sleep researchers for running the sleep video analysis on their videos.

We considered the problem of estimating the approximate location of a web cam by analyzing its images. We showed that we could effectively estimate locations with less than 2.4% error for the longitude and less than 5% error for the latitude.

The main contributions of this thesis are listed as follows:

- We improved VHR for assessing resting HR in a controlled setting where the subject has no motion. We modified and extend an ICA-based method and improve its performance by (1) adapting the passband of the bandpass filter (BPF) or the temporal filter, (2) by removing background noise from the signal by matching and removing signals that occur in the off-target (background) and on-target areas (facial region), and (3) detect skin pixels within the facial region to exclude pixels that does not contain HR signal.
- We investigated and described the motion effects in VHR in terms of the angle change of the subject’s skin surface in relation to the light source. We showed that the illumination change on each surface point is one of the major factors causing motion artifacts by estimating the incident angle in each frame. Based on this understanding, we discussed the future work on how we can compensate for the motion artifacts.
- We proposed auto-VSG method where we used child head size to normalize the motion index and to provide an individual motion maximum for each child. We compared the proposed auto-VSG method to (1) traditional B-VSG sleep-awake labels and (2) actigraphy sleep vs. wake estimates across four sleep parameters: sleep onset time, sleep offset time, awake duration, and sleep duration. In sum, analyses revealed that estimates generated from the proposed auto-VSG method and B-VSG are comparable.
- In the next proposed auto-VSG method, we described an automated VSG sleep detection system which uses deep learning approaches to label frames in a sleep video as “sleep” or “awake” in young children. We examined 3D Convolutional Networks (C3D) and Long Short-term Memory (LSTM) relative to motion information from selected Groups of Pictures of a sleep video and tested temporal window sizes for back propagation. We compared our proposed VSG methods to traditional B-VSG sleep-awake labels. C3D had an accuracy of approximately 90% and the proposed LSTM method improved the accuracy to more than 95%.

The analyses revealed that estimates generated from the proposed LSTM-based method with long-term temporal dependency are suitable for automated sleep or awake labeling.

- We created web application (Sleep Web App) that makes our sleep analysis methods accessible to run from web browsers regardless of users' working environments. The design philosophy of Sleep Web App is to serve easy accesses to sleep researchers for running the sleep video analysis on their videos. Specifically, we focused on (1) simple user experience, (2) multi-user supporting and (3) providing results for further analysis. For providing the results, we included two csv format files for per-minute sleep analysis and sleep summary results.
- We also described a method for estimating the location of an IP-connected camera (a web cam) by analyzing a sequence of images obtained from the camera. First, we classified each image as Day/Night using the mean luminance of the sky region. From the Day/Night images, we estimated the sunrise/set, the length of the day, and local noon. Finally, the geographical location (latitude and longitude) of the camera is estimated. The experiment results show that our approach achieves reasonable performance.

## 7.2 Future Work

To extend our work on video-based HR estimation, known as videoplethysmography (VHR), to more general cases that can cover various recording scenarios, there are some future work to be done. In Chapter 2, we adapted an existing video-based HR estimation method to produce more robust and accurate results. However, the method works poor when the subject is moving during the recording. In Chapter 3.1, we showed that the linearity assumption used in conventional HR estimation methods no longer hold when there is subject motions in the video. To understand this motion effects in VHR, we showed the relationship between the motion and the intensity change by setting up two experiments. Our experiments showed how the incident



angle change caused by motion is related to the pixel intensity changes. We showed that the illumination change on each surface point is one of the major factors causing motion artifacts. In Chapter 3.5, we provided initial work on how motion effects could be estimated as  $L(\hat{n})$  using the facial landmark tracking and approximate lighting directions on some test videos. To extend this  $L(\hat{n})$  estimation to more general scenarios, following are suggested:

1. improving the tracking performance of three facial points,
2. instead of using fixed values for all the frames, the light source direction for each frame could be estimated using the location and shadow information and
3. a method to find sub-region (surface) that share the same surface normal ( $\vec{n}_k(n)$ , described in Chapter 3.5) could be investigated so that more surface normal estimation can be done more accurately.

Once we have method for  $L(\hat{n})$  estimation, another future work to be done for motion-robust VHR is non-linear filtering method. A method to filter out PPG signal from the actual intensity change  $L(n)$ , that includes both motion effects and PPG signal, should be further investigated. Details are described in Chapter 3.4.

Our study includes VHR experiments for specific motion, periodically moving from side to side. With more work on estimating motion effects from videos and devising filtering methods, the work can be extended to VHR for various different motions.

### 7.3 Publications Resulting From This Thesis

1. **J. Choe**, A. J. Schwichtenberg, E. J. Delp, “Classification of Sleep Videos Using Deep Learning,” *Proceedings of the IEEE Multimedia Information Processing and Retrieval*, pp. 115–120, March 2019, San Jose, CA.

2. A. J. Schwichtenberg, **J. Choe**, A. Kellerman, E. Abel and E. J. Delp, "Pediatric Videosomnography: Can signal/video processing distinguish sleep and wake states?," *Frontiers in Pediatrics*, vol. 6, num. 158, pp. 1-11, May 2018.
3. **J. Choe**, D. Mas Montserrat, A. J. Schwichtenberg and E. J. Delp, "Sleep Analysis Using Motion and Head Detection," *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 29-32, April 2018, Las Vegas, NV.
4. D. Chung, **J. Choe**, M. O'Haire, A. J. Schwichtenberg and E. J. Delp, "Improving Video-Based Heart Rate Estimation," *Proceedings of the Electronic Imaging, Computational Imaging XIV*, pp. 1-6(6), February, 2016, San Francisco, CA.
5. **J. Choe**, D. Chung, A. J. Schwichtenberg, and E. J. Delp, "Improving video-based resting heart rate estimation: A comparison of two methods," *Proceedings of the IEEE 58th International Midwest Symposium on Circuits and Systems*, pp. 1-4, August 2015, Fort Collins, CO.
6. T. Pramoun, **J. Choe**, H. Li, Q. Chen, T. Amornraksa, Y. Lu, and E. J. Delp, "Webcam classification using simple features," *Proceedings of the SPIE/IS&T International Symposium on Electronic Imaging*, pp. 94010G:1-12, March 2015, San Francisco, CA.
7. **J. Choe**, T. Pramoun, T. Amornraksa, Y. Lu, and E. J. Delp, "Image-based geographical location estimation using web cameras," *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 73-76, April 2014, San Diego, CA.

## REFERENCES

## REFERENCES

- [1] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, “Wearable photoplethysmographic sensors—past and present,” *Electronics*, vol. 3, no. 2, pp. 282–302, April 2014.
- [2] O. S. Ipsiroglu, Y. A. Hung, F. Chan, M. L. Ross, D. Veer, S. Soo, G. Ho, M. Berger, G. McAllister, H. Garn, G. Kloesch, A. V. Barbosa, S. Stockler, W. McKellin, and E. Vatikiotis-Bateson, ““diagnosis by behavioral observation” home-videosomnography ? a rigorous ethnographic approach to sleep of children with neurodevelopmental conditions,” *Front Psychiatry*, vol. 6, no. 39, pp. 1–15, March 2015.
- [3] A. Sadeh, “III. sleep assessment methods,” *Monographs of the Society for Research in Child Development*, vol. 80, no. 1, pp. 33–48, February 2015.
- [4] “Pulse,” URL: <https://www.nlm.nih.gov/medlineplus/>.
- [5] Y. Ostachega, K. Porter, J. Hughes, C. Dillon, and T. Nwankwo, “Resting pulse rate reference data for children, adolescents, and adults: United states, 1999–2008,” *National Health Statistics Reports*, no. 41, pp. 1–16, August 2011.
- [6] J. Allen, “Photoplethysmography and its application in clinical physiological measurement,” *Physiological Measurement*, vol. 28, no. 3, pp. R1–R39, March 2007.
- [7] L. G. Lindberg, T. Tamura, and P. A. Oberg, “Photoplethysmography,” *Physiological Measurement*, vol. 29, no. 1, pp. 40–47, January 1991.
- [8] K. H. Shelley, “Photoplethysmography: Beyond the calculation of arterial oxygen saturation and heart rate,” *Anesthesia & Analgesia*, vol. 105, no. 6, pp. S31–S36, December 2007.
- [9] A. A. Kamal, J. B. Harness, G. Irving, and A. J. Mearns, “Skin photoplethysmography—a review,” *Computer Methods and Programs in Biomedicine*, vol. 28, no. 4, pp. 257–69, April 1989.
- [10] A. V. J. Challoner and C. A. Ramsay, “Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm,” *Physics in Medicine and Biology*, vol. 19, no. 3, pp. 317–328, October 1973.
- [11] M. Elgendi, “On the analysis of fingertip photoplethysmogram signals,” *Current Cardiology Reviews*, vol. 8, no. 1, pp. 14–25, February 2012.
- [12] R. Ortega, C. Hansen, K. Elterman, and A. Woo, “Pulse oximetry,” *The New England Journal of Medicine*, vol. 364, no. 16, p. e33, 2011.

- [13] “American Heart Association,” URL: <http://www.heart.org/HEARTORG/>.
- [14] M. Bolanos, H. Nazeran, and E. Haltiwanger, “Comparison of heart rate variability signal features derived from electrocardiography and photoplethysmography in healthy individuals,” *Proceedings of the 28th IEEE Annual International Conference on Engineering in Medicine and Biology Society*, pp. 4289–4294, August 2006, new York, NY.
- [15] G. Lu, F. Yang, J. A. Taylor, and J. F. Stein, “A comparison of photoplethysmography and ecg recording to analyse heart rate variability in healthy subjects,” *Journal of Medical Engineering & Technology*, vol. 33, no. 8, pp. 634–641, December 2009.
- [16] M. Poh, K. Kim, A. Goessling, N. Swenson, and R. Picard, “Cardiovascular monitoring using earphones and a mobile device,” *IEEE Pervasive Computing*, vol. 11, no. 4, pp. 18–26, 2012.
- [17] D. Grimaldi, Y. Kurylyak, F. Lamonaca, and A. Nastro, “Photoplethysmography detection by smartphone’s videocamera,” *Proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, pp. 488–491, September 2011, prague, Czech Republic.
- [18] C. G. Scully, J. Lee, J. Meyer, A. M. Gorbach, D. Granquist-Fraser, Y. Mendelson, and K. H. Chon, “Physiological parameter monitoring from optical recordings with a mobile phone,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 303 – 306, 2011.
- [19] E. Pinheiro, O. Postolache, and P. Gir?o, “Theory and developments in an unobtrusive cardiovascular system representation: Ballistocardiography,” *The Open Biomedical Engineering Journal*, vol. 4, p. 201?216, October 2010.
- [20] J. Paalasmaa, H. Toivonen, and M. Partinen, “Adaptive heartbeat modeling for beat-to-beat heart rate measurement in ballistocardiograms,” *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1945–1952, 2015.
- [21] J. Hernandez, Y. Li, J. M. Rehg, and R. W. Picard, “Bioglass: Physiological parameter estimation using a head-mounted wearable device,” *Proceedings of the 2014 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth)*, pp. 55–58, November 2014, Athens, Greece.
- [22] R. Gonzalez-Landaeta, O. Casas, and R. Pallas-Areny, “Heart rate detection from plantar bioimpedance measurements,” *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1163 – 1167, February 2008.
- [23] I. Mikhelson, P. Lee, S. Bakhtiari, T. Elmer, A. Katsaggelos, and A. Sahakian, “Noncontact millimeter-wave real-time detection and tracking of heart rate on an ambulatory subject,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 5, pp. 927–934, September 2012.
- [24] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, “Smart homes that monitor breathing and heart rate,” *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 837–846, April 2015, seoul, Korea.

- [25] C. Takano and Y. Ohta, "Heart rate measurement based on a time-lapse image," *Medical Engineering & Physics*, vol. 29, no. 8, pp. 853–857, October 2007.
- [26] W. Verkruyse, L. O. Svaasand, and J. S. Nelson, "Remote plethysmographic imaging using ambient light," *Optics Express*, vol. 16, no. 26, pp. 21 434–21 445, December 2008.
- [27] M. Poh, D. McDuff, and R. Picard, "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation," *Optics Express*, vol. 18, no. 10, pp. 10 762–10 774, May 2010.
- [28] M. Poh, D. McDuff, and R. Picard, "Advancements in noncontact, multiparameter physiological measurements using a webcam," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 7–11, January 2011.
- [29] D. McDuff, S. Gontarek, and R. Picard, "Improvements in remote cardiopulmonary measurement using a five band digital camera," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2593 – 2601, October 2014.
- [30] H. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 65:1–8, July 2010.
- [31] F. Zhao, M. Li, Y. Qian, and J. Tsien, "Remote measurements of heart and respiration rates for telemedicine," *PLoS ONE*, vol. 8, no. 10, October 2013, e71384.
- [32] X. Li, J. Chen, G. Zhao, and M. Pietikainen, "Remote heart rate measurement from face videos under realistic situations," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4264–4271, June 2014, Columbus, OH.
- [33] G. Balakrishnan, F. Durand, and J. Guttag, "Detecting pulse from head motions in video," *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3430–3437, June 2013, Portland, OR.
- [34] H. K. S. Kwon and K. Park, "Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2174–2177, September 2012, San Diego, CA.
- [35] M. Kumar, A. Veeraraghavan, and A. Sabharwal, "DistancePPG: Robust non-contact vital signs monitoring using a camera," *Biomedical Optics Express*, vol. 6, no. 5, pp. 1565–1588, 2015.
- [36] A. Lam and Y. Kuno, "Robust heart rate measurement from video using select random patches," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3640–3648, December 2015, Santiago, Chile.
- [37] P. Sahindrakar, "Improving motion robustness of contact-less monitoring of heart rate using video analysis," Ph.D. dissertation, Technische Universiteit Eindhoven, Department of Mathematics and Computer Science, 2011.

- [38] Y. Sun, S. Hu, V. Azorin-Peris, S. Greenwald, J. Chambers, and Y. Zhu, "Motion-compensated noncontact imaging photoplethysmography to monitor cardiorespiratory status during exercise," *Journal of Biomedical Optics*, vol. 16, no. 7, pp. 077010:1–9, July 2011.
- [39] Y. Sun, C. Papin, V. Azorin-Peris, R. Kalawsky, S. Greenwald, and S. Hua, "Use of ambient light in remote photoplethysmographic systems: comparison between a high-performance camera and a low-cost webcam," *Journal of Biomedical Optics*, vol. 17, no. 3, pp. 037005:1–10, March 2012.
- [40] G. R. Tsouri, S. Kyal, S. Dianat, and L. K. Mestha, "Constrained independent component analysis approach to nonobtrusive pulse rate measurements," *Journal of Biomedical Optics*, vol. 17, no. 7, pp. 077011:1–4, July 2012.
- [41] J. R. Estepp, E. B. Blackford, and C. M. Meier, "Recovering pulse rate during motion artifact with a multi-imager array for non-contact imaging photoplethysmography," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1462–1469, October 2014, San Diego, CA.
- [42] Y. Yu, P. Raveendran, and C. Lim, "Dynamic heart rate measurements from video sequences," *Biomedical Optics Express*, vol. 6, no. 7, pp. 2466–2480, 2015.
- [43] D.-Y. Chen, J.-J. Wang, K.-Y. Lin, H.-H. Chang, H.-K. Wu, Y.-S. Chen, and S.-Y. Lee, "Image sensor-based heart rate evaluation from face reflectance using hilbert-huang transform," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 618–627, January 2015.
- [44] A. G. Garcia, "Development of a non-contact heart rate measurement system," Master's thesis, University of Edinburgh, School of Informatics, 2013.
- [45] M. Lewandowska, J. Ruminski, T. Kocejko, and J. Nowak, "Measuring pulse rate with a webcam—a non-contact method for evaluating cardiac activity," *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp. 405–410, September 2011, Szczecin, Poland.
- [46] Y. Yu, P. Raveendran, C. Lim, and B. Kwan, "Dynamic heart rate estimation using principal component analysis," *Biomedical Optics Express*, vol. 6, no. 11, pp. 4610–4618, November 2015.
- [47] D. N. Tran, H. Lee, and C. Kim, "A robust real time system for remote heart rate measurement via camera," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1–6, June 2015, Turin, Italy.
- [48] L. Wei, Y. Tian, Y. Wang, T. Ebrahimi, and T. Huang, "Automatic webcam-based human heart rate measurements using laplacian eigenmap," *Proceedings of the 11th Asian conference on Computer Vision*, pp. 281–292, November 2012, Daejeon, Korea.
- [49] A. Zhao, F. Durand, and J. Guttag, "Estimating a small signal in the presence of large noise," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 420–25, December 2015, Santiago, Chile.
- [50] U. Bal, "Non-contact estimation of heart rate and oxygen saturation using ambient light," *Biomedical Optics Express*, vol. 6, no. 1, pp. 86–97, Jan 2015.

- [51] J. Gunther, N. Ruben, and T. Moon, "Model-based (passive) heart rate estimation using remote video recording of moving human subjects illuminated by ambient light," *Proceedings of the IEEE International Conference on Image Processing*, 2015.
- [52] L. Tarassenko, M. Villarroel, A. Guazzi, J. Jorge, D. A. Clifton, and C. Pugh, "Non-contact video-based vital sign monitoring using ambient light and autoregressive models," *Physiological Measurement*, vol. 35, no. 5, pp. 807–831, March 2014.
- [53] S. Yu, X. You, X. Jiang, K. Zhao, Y. Mou, W. Ou, Y. Tang, and C. L. P. Chen, "Human heart rate estimation using ordinary cameras under natural movement," *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1041–1046, October 2015, Kowloon, China.
- [54] L. Feng, L. Po, X. Xu, Y. Li, and R. Ma, "Motion-resistant remote imaging photoplethysmography based on the optical properties of skin," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 879–891, 2015.
- [55] L. Feng, L. Po, X. Xu, Y. Li, C. Cheung, K. Cheung, and F. Yuan, "Dynamic roi based on k-means for remote photoplethysmography," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1310–1314, April 2015, south Brisbane, Australia.
- [56] Y. Yu, K. Lumpur, R. Paramesran, and C. Lim, "Video based heart rate estimation under different light illumination intensities," *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 216 – 221, December 2014, Kuching, Malaysia.
- [57] G. Haan and V. Jeanne, "Robust pulse rate from chrominance-based rppg," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2878–2886, 2013.
- [58] G. Haan and A. Leest, "Improved motion robustness of remote-ppg?by using the blood volume pulse signature," *Physiological Measurement*, vol. 35, no. 9, pp. 1913–1926, 2014.
- [59] W. Wang, S. Stuijk, and G. Haan, "Exploiting spatial redundancy of image sensor for motion robust rPPG," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 415–425, January 2015.
- [60] M. van Gastel, S. Stuijk, and G. Haan, "Motion robust remote-ppg in infrared," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 5, pp. 1425–1433, May 2015.
- [61] A. V. Moco, S. Stuijk, and G. Haan, "Ballistocardiographic artifacts in ppg imaging," *IEEE Transactions on Biomedical Engineering*, vol. PP, no. 99, pp. 1–8, November 2015.
- [62] H. Monkaresi, R. Calvo, and H. Yan, "A machine learning approach to improve contactless heart rate monitoring using a webcam," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1153–1160, November 2013.



- [63] Y. Yan, X. Ma, L. Yao, and J. Ouyang, "Noncontact measurement of heart rate using facial video illuminated under natural light and signal weighted analysis," *Bio-Medical Materials and Engineering*, vol. 26, no. s1, pp. 903–909, 2015.
- [64] S. Xu, L. Sun, and G. K. Rohde, "Robust efficient estimation of heart rate pulse from video," *Biomedical Optics Express*, vol. 5, no. 4, pp. 1124–1135, April 2014.
- [65] P. Werner, A. Al-Hamadi, S. Walter, S. Gruss, and H. C. Harald, "Automatic heart rate estimation from painful faces," *Proceedings of the IEEE International Conference on Image Processing*, pp. 1947–1951, 2014.
- [66] H. Tasli, A. Gudi, and M. Uyl, "Remote ppg based vital sign measurement using adaptive facial regions," *Proceedings of the IEEE International Conference on Image Processing*, pp. 1410–1414, Oct 2014.
- [67] R. Stricker, S. Muller, and H. Gross, "Non-contact video-based pulse rate measurement on a mobile service robot," *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 1056 – 1062, August 2014, edinburgh, Scotland.
- [68] G. Cennini, J. Arguel, K. Ak?it, and A. van Leest, "Heart rate monitoring via remote photoplethysmography with motion artifacts reduction," *Optics Express*, vol. 18, no. 5, pp. 4867–4875, 2010.
- [69] R. Amelard, C. Scharfenberger, F. Kazemzadeh, K. J. Pfisterer, B. S. Lin, D. A. Clausi, and A. Wong, "Feasibility of long-distance heart rate monitoring using transmittance photoplethysmographic imaging," *Scientific Reports*, vol. 5, no. 14637, pp. 1–11, October 2015.
- [70] H. Qi, Z. J. Wang, and C. Miao, "Non-contact driver cardiac physiological monitoring using video data," *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing*, pp. 418 – 422, July 2015, Chengdu, China.
- [71] R. Huang and L. Dung, "A motion-robust contactless photoplethysmography using chrominance and adaptive filtering," *Proceedings of the IEEE Biomedical Circuits and Systems Conference*, pp. 1–4, October 2015, Atlanta, GA.
- [72] A. Shagholi, M. Charmi, and H. Rakhshan, "The effect of the distance from the webcam in heart rate estimation from face video images," *Proceedings of the 2nd International Conference on Pattern Recognition and Image Analysis*, pp. 1–6, March 2015, Rasht, Iran.
- [73] D. Lee, J. Kim, S. Kwon, and K. Park, "Heart rate estimation from facial photoplethysmography during dynamic illuminance changes," *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2758 – 2761, August 2015, Milan, Italy.
- [74] A. M. Rodr?guez and J. R. Castro, "Pulse rate variability analysis by video using face detection and tracking algorithms," *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5696 – 5699, August 2015, Milan, Italy.

- [75] K. Lin, D. Chen, and W. Tsai, "Face-based heart rate signal decomposition and evaluation using multiple linear regression," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1351 – 1360, March 2016.
- [76] C. Huang, X. Yang, and K. Cheng, "Accurate and efficient pulse measurement from facial videos on smartphones," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, p. To appear, March 2016, Lake Placid, NY.
- [77] U. S. Freitas, "Remote camera-based pulse oximetry," *Proceedings of the 6th International Conference on eHealth, Telemedicine, and Social Medicine*, pp. 59–63, March 2014, Barcelona, Spain.
- [78] H. Pan, D. Temel, and G. AlRegib, "Heartbeat: Heart beat estimation through adaptive tracking," *Proceedings of the IEEE International Conference on Biomedical and Health Informatics*, pp. 587–590, February 2016, Las Vegas, NV.
- [79] J. Deglint, A. G. Chung, B. Chwyl, R. Amelard, F. Kazemzadeh, X. Y. Wang, D. A. Clausi, and A. Wong, "Photoplethysmographic imaging via spectrally demultiplexed erythema fluctuation analysis for remote heart rate monitoring," *Proceedings of the SPIE Multimodal Biomedical Imaging XI, 970111*, pp. 1–6, February 2016, San Francisco, CA.
- [80] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434 – 444, February 1997.
- [81] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, June 2000.
- [82] S. H. Fouladi, I. B., T. A. Ramstad, and K. Kansanen, "Accurate heart rate estimation from camera recording via music algorithm," *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 7454 – 7457, August 2015, Milan, Italy.
- [83] D. A. Forsyth and J. Ponce, *Computer Vision, A Modern Approach*. Upper Saddle River, NJ: Pearson Education, Inc., 2003, vol. 1.
- [84] "Computer Vision Lecture Notes by Avinash Kak," URL: <https://engineering.purdue.edu/kak/computervision/>.
- [85] I. O. Kirenko, G. Haan, A. J. V. Leest, and R. S. Mulyar, "Video coding and decoding devices and methods preserving ppg relevant information," Patent US 2013/0 272 393 A1, October 17, 2013.
- [86] M. Raghuram, K. Madhav, E. Krishna, and K. Reddy, "Evaluation of wavelets for reduction of motion artifacts in photoplethysmographic signals," *Proceedings of the 10th International Conference on Information Sciences Signal Processing and their Applications*, pp. 450–463, May 2010.
- [87] R. W. C. G. R. Wijshoff, M. Mischi, P. H. Woerlee, and R. M. Aarts, "Improving pulse oximetry accuracy by removing motion artifacts from photoplethysmograms using relative sensor motion: A preliminary study," in *Oxygen Transport to Tissue XXXV*, S. V. Huffel, G. Naulaers, A. Caicedo, D. F. Bruley, and D. K. Harrison, Eds. NY: Springer, 2013, vol. 789, pp. 411–417.

- [88] J. Lee, K. Matsumura, K. Yamakoshi, P. Rolfe, S. Tanaka, and T. Yamakoshi, "Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion," *Proceedings of the IEEE 35th Annual International Conference on Engineering in Medicine and Biology Society*, pp. 1724 – 1727, July 2013, Osaka, Japan.
- [89] Z. Zhang, Z. Pi, and B. Liu, "TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522–531, Feb 2015.
- [90] M. J. Hayes, "Artefact reduction in photoplethysmography," Ph.D. dissertation, Loughborough University, Department of Electronic and Electrical Engineering, 1998.
- [91] H. Werner, L. Molinari, C. Guyer, and O. G. Jenni, "Agreement rates between actigraphy, diary, and questionnaire for children's sleep patterns," *Archives of Pediatrics & Adolescent Medicine*, vol. 162, no. 4, pp. 350–358, April 2008.
- [92] A. Sadeh, "The role and validity of actigraphy in sleep medicine: An update," *Sleep Medicine Reviews*, vol. 15, no. 4, pp. 259–267, August 2011.
- [93] S. Okada, Y. Ohno, Goyahan, K. Kato-Nishimura, I. Mohri, and M. Tanike, "Examination of non-restrictive and non-invasive sleep evaluation technique for children using difference images," *Proceedings of the IEEE 30th Annual International Conference on Engineering in Medicine and Biology Society*, pp. 3483–3487, August 2008, Vancouver, BC.
- [94] M. Nakatani, S. Okada, S. Shimizu, I. Mohri, Y. Ohno, M. Taniike, and M. Makikawa, "Body movement analysis during sleep for children with adhd using video image processing," *Proceedings of the IEEE 35th Annual International Conference on Engineering in Medicine and Biology Society*, pp. 6389–6392, July 2013, Osaka, Japan.
- [95] A. Heinrich, X. Aubert, and G. Haan, "Body movement analysis during sleep based on video motion estimation," *Proceedings of the IEEE 15th International Conference on e-Health Networking, Applications and Services*, pp. 539–543, October 2013, Lisbon, Portugal.
- [96] S. Okada and M. M. N. Shiozawa, "Body movement in children with adhd calculated using video images," *Proceedings of the IEEE 35th Annual International Conference on Engineering in Medicine and Biology Society*, pp. 60–61, January 2012, Hong Kong, China.
- [97] L. Atzoria, A. Ierab, and G. Morabitoc, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, October 2010.
- [98] D. Miorandi, S. Sicari, F. Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, September 2012.
- [99] "ITU Internet Reports 2005: The Internet of Things," *International Telecommunication Union Technical Report*, November 2005.

- [100] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, September 2013.
- [101] B. Guo, D. Zhang, and Z. Wang, "Living with Internet of Things: The emergence of embedded intelligence," *Proceedings of the Internet of Things, 4th International Conference on Cyber, Physical and Social Computing*, pp. 297–304, October 2011, Dalian, China.
- [102] A. S. Kaseb, E. Berry, Y. Koh, A. Mohan, W. Chen, H. Li, Y.-H. Lu, and E. J. Delp, "A system for large-scale analysis of distributed cameras," *Proceedings of the IEEE Global Conference on Signal and Information Processing*, pp. 340 – 344, December 2014, atlanta, GA.
- [103] A. S. Kaseb, W. Chen, G. Gingade, and Y.-H. Lu, "Worldview and route planning using live public cameras," *Proceedings of the SPIE/IS&T Electronic Imaging, Imaging and Multimedia Analytics in a Web and Mobile World*, pp. 1–8, March 2015, san Francisco, CA.
- [104] A. S. Kaseb, E. Berry, E. Rozolis, K. McNulty, S. Bontrager, Y. Koh, Y.-H. Lu, and E. J. Delp, "An interactive web-based system for large-scale analysis of distributed cameras," *Proceedings of the SPIE/IS&T Electronic Imaging, Imaging and Multimedia Analytics in a Web and Mobile World*, pp. 1–11, March 2015, san Francisco, CA.
- [105] T. J. Hacker and Y.-H. Lu, "An instructional cloud-based testbed for image and video analytics," *Proceedings of the IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 859 – 862, December 2014, singapore.
- [106] "Latitude and Longitude," URL: <http://nationalatlas.gov/>.
- [107] H. Read and J. Watson, *Introduction to Geology*. New York: Halsted, 1975, pp. 13–15.
- [108] D. Sobel, *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*, 10th ed. Walker & Company, 2007.
- [109] P. K. Seidelmann, *Explanatory Supplement to the Astronomical Almanac*. University Science Books, 2005, pp. 32–33.
- [110] W. Forsythe, E. Rykiel Jr., R. Stahla, H. Wua, and R. Schoolfield, "A model comparison for daylength as a function of latitude and day of year," *Ecological Modelling*, vol. 80, no. 1, pp. 87–95, June 1995.
- [111] J. Hays and A. A. Efros, "Im2gps: estimating geographic information from a single image," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8, June 2008, anchorage, AK.
- [112] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister, "What do color changes reveal about an outdoor scene?" *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8, June 2008, anchorage, AK.

- [113] J. Lalonde, S. G. Narasimhan, and A. A. Efros, “What do the sun and the sky tell us about the camera?” *International Journal of Computer Vision*, vol. 88, no. 1, pp. 24–51, September 2009.
- [114] I. N. Junejo and H. Foroosh, “Estimating geo-temporal location of stationary cameras using shadow trajectories,” *Proceedings of the 10th European Conference on Computer Vision*, pp. 318–331, October 2008, marseille, France.
- [115] L. Wu and X. Cao, “Geo-location estimation from two shadow trajectories,” *Proceedings of the IEEE*, pp. 585 – 590, June 2010, san Francisco, CA.
- [116] F. Sandnes, “A simple content-based strategy for estimating the geographical location of a webcam,” *Proceedings of the 11th Pacific Rim Conference on Multimedia*, vol. 6297, pp. 36–45, September 2010, Shanghai, China.
- [117] N. Laungrungthip, A. E. McKinnon, C. D. Churcher, and K. Unsworth, “Edge-based detection of sky regions in images for solar exposure prediction,” *Proceedings of the 23rd International Conference on Image and Vision Computing New Zealand*, pp. 1–6, November 2008, Christchurch, New Zealand.
- [118] S. Kim, S. Oh, J. Kang, Y. Ryu, K. Kim, S. Park, and K. Park, “Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2173–2178, August 2005, Alberta, Canada.
- [119] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea, and M. Vetterli, “How is the weather: Automatic inference from images,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 1853–1856, September 2012, Orlando, FL.
- [120] M. Tarvainen, P. Ranta-aho, and P. Karjalainen, “An advanced detrending method with application to hrv analysis,” *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 2, pp. 172–175, February 2002.
- [121] J. Vila, F. Palacios, J. Presedo, M. Fernandez-Delgado, P. Felix, and S. Barro, “Time-frequency analysis of heart-rate variability,” *IEEE Magazine on Engineering in Medicine and Biology*, vol. 16, no. 5, pp. 119–126, Sep 1997.
- [122] N. Wadhwa, M. Rubinstein, F. Durand, and W. Freeman, “Phase-based video motion processing,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 80:1–9, July 2013.
- [123] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I–511–I–518, December 2001, Kauai, HI.
- [124] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” *Proceedings of the 7th European Conference on Computer Vision*, pp. 661–675, May 2002, Copenhagen, Denmark.
- [125] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

- [126] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 274–280, June 1999, Fort Collins, CO.
- [127] A. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ: Prentice Hall, 1989, vol. 1.
- [128] Q. Zhu and Y. W. C. Wu, K. Cheng, "An adaptive skin model and its application to objectionable image filtering," *Proceedings of the ACM International Conference on Multimedia*, pp. 56–63, October 2004, New York, NY.
- [129] Pantone, *Pantone Skin Tone Guide*. Carlstadt, NJ: Pantone Inc., 2012.
- [130] J. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.
- [131] J. Choe, D. Chung, A. J. Schwichtenberg, and E. J. Delp, "Improving video-based resting heart rate estimation: A comparison of two methods," *Proceedings of the IEEE 58th International Midwest Symposium on Circuits and Systems*, pp. 1–4, August 2015, Fort Collins, CO.
- [132] D. Chung, J. Choe, M. E. O’Haire, A. Schwichtenberg, and E. J. Delp, "Improving video-based heart rate estimation," *Proceedings of the IS&T International Symposium on Electronic Imaging*, p. To appear, February 2016, San Francisco, CA.
- [133] H. Barrow and J. Tannenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems (A. Hanson and E. Riseman (Eds.))*, no. 157, pp. 3–26, 1978.
- [134] J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, pp. 1–9, 2000.
- [135] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters*. Norwell, MA: Kluwer Academic Publishers, 1990, vol. 1.
- [136] O. R. Mitchell, E. J. Delp, and P. L. Chen, "Filtering to remove cloud cover in satellite imagery," *IEEE Transactions on Geoscience Electronics*, vol. GE-15, no. 3, pp. 137–141, July 1977.
- [137] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY: Cambridge University Press, 2004, vol. 1.
- [138] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2729 – 2736, June 2010, San Francisco, CA.
- [139] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476 – 3483, June 2013, Portland, OR.
- [140] Y. Tie and L. Guan, "Automatic landmark point detection and tracking for human facial expressions," *Journal on Image and Video Processing*, vol. 2013, no. 8, pp. 1–15, February 2013.

- [141] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867 – 1874, June 2014, Columbus, OH.
- [142] N. Markus, M. Frljak, I. S. Pandzic, J. Ahlberg, and R. Forchheimer, "Fast localization of facial landmark points," *Technical Report*, January 2015, University of Zagreb, Zagreb, Croatia.
- [143] I. Moon, K. Kim, J. Ryu, and M. Mun, "Face direction-based human-computer interface using image observation and emg signal for the disabled," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1515 – 1520, September 2003, Taipei, Taiwan.
- [144] Z. Zhu and Q. Ji, "3d face pose tracking from an uncalibrated monocular camera," *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 400 – 403, August 2004, England, UK.
- [145] Y. Matsumoto, J. Ido, K. Takemura, M. Koeda, and T. Ogasawara, "Portable facial information measurement system and its application to human modeling and human interfaces," *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 475 – 480, May 2004, Seoul, Korea.
- [146] P. Smith, M. Shah, and N. Lobo, "Determining driver visual attention with one camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 205 – 218, December 2003.
- [147] X. Wang, H. Huang, Z. Ruan, and Z. Lu, "Fast face orientation estimation from an uncalibrated monocular camera," *Proceedings of the Congress on Image and Signal Processing*, pp. 186 – 190, May 2008, Sanya, China.
- [148] A. Sadeh, K. M. Sharkey, and M. A. Carskadon, "Activity-based sleep-wake identification: An empirical test of methodological issues," *SLEEP*, vol. 17, no. 3, pp. 201–207, April 1994.
- [149] S. Ancoli-Israel, R. Cole, C. Alessi, M. Chambers, W. Moorcroft, and C. Pollak, "The role of actigraphy in the study of sleep and circadian rhythms," *SLEEP*, vol. 26, no. 3, pp. 342–392, May 2003.
- [150] M. Piccardi, "Background subtraction techniques: a review," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, October 2004.
- [151] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014, Columbus, OH.
- [152] R. Girshick, "Fast R-CNN," *Proceedings of the International Conference on Computer Vision*, pp. 1440–1448, December 2015, Santiago, Chile.
- [153] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 91–99, December 2015, Montreal, Canada.

- [154] T. Vu, A. Osokin, and I. Laptev, "Context-aware CNNs for person head detection," *Proceedings of the International Conference on Computer Vision*, pp. 2893–2901, December 2015, Santiago, Chile.
- [155] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Proceedings of the European Conference on Computer Vision*, pp. 818–833, September 2014, Zürich, Switzerland.
- [156] A. J. Schwichtenberg, T. Hensle, S. Honaker, M. Miller, S. Ozonoff, and T. Anders, "Sibling sleep - what can it tell us about parental sleep reports in the context of autism?" *Clinical Practice in Pediatric Psychology*, vol. 4, no. 2, pp. 137–152, June 2016.
- [157] M. Moore, V. Evans, G. Hanvey, and C. Johnson, "Assessment of sleep in children with autism spectrum disorder," *Children (Basel)*, vol. 4, no. 72, pp. 1–17, August 2017.
- [158] D. Hodge, A. M. Parnell, C. D. Hoffman, and D. P. Sweeney, "Methods for assessing sleep in children with autism spectrum disorders: A review," *Research in Autism Spectrum Disorders*, vol. 6, no. 4, pp. 1337–1344, October 2012.
- [159] Sleep Research Society, *Basics of Sleep Behavior*. Edinburgh, UK: UCLA, 1993.
- [160] W. Liao and C. Yang, "Video-based activity and movement pattern analysis in overnight sleep studies," *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 1–4, December 2008, tampa, FL.
- [161] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: explorations in the microstructure of cognition*, D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, pp. 318–362.
- [162] A. Graves, "Supervised sequence labelling with recurrent neural networks," vol. 385, 2012.
- [163] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [164] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [165] A. Graves, "Generating sequences with recurrent neural networks," *arXiv:1308.0850v5*, pp. 1–43, June 2014.
- [166] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097–1105, December 2012.
- [167] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, June 2014, columbus, OH.



- [168] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” 2014.
- [169] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.
- [170] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4489–4497, December 2015, santiago, Chile.
- [171] “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” *CRCV-TR-12-01*, 2012, University of Central Florida, Orlando, FL.
- [172] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, July 2011.
- [173] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.
- [174] A. Nielsen, K. Bigelow, M. Musyl, and J. Sibert, “Improving light-based geolocation by including sea surface temperature,” *Fisheries Oceanography*, vol. 15, no. 4, pp. 314–325, July 2006.
- [175] “Federal Aviation Administration,” URL: <http://akweathercams.faa.gov/>.
- [176] “OpenCV (Open Source Computer Vision Library),” URL: <https://opencv.org/>.
- [177] “TensorFlow,” URL: <https://www.tensorflow.org/>.
- [178] “Apache2 Documentation,” URL: <http://httpd.apache.org/docs/2.4/>.
- [179] “flask,” URL: <http://flask.pocoo.org/>.

## APPENDICES

## A. SLEEP WEB APPLICATION

### A.1 Introduction

The sleep project is collaboration work with Dr. Schwichtenberg's Sleep and Development Lab. To share sleep analysis methods described in Chapter 5 with sleep researchers, we created a web application that makes our sleep analysis methods accessible to run from web browsers regardless of users' work environment. We call this application Sleep Web App.

Sleep Web App lets users classify their videos using the sleep/awake classification method described in Chapter 5.

The design purpose of Sleep Web App is to provide easy accesses for sleep researchers to apply auto-VSG to existing videos. Specifically, we focused on following three things.

First is simple user experience. After uploading videos in a zip archive, the server automatically runs sleep/awake classification. When the processing is done, the result table is displayed in the web browser. Users do not need to do any installation. Since the program runs in the server, specifications of the user's computers do not matter as long as they can access web browsers. Additionally, users do not need to worry about the maintenance of the program.

Second is multi-user supporting. Sleep Web App is designed to process multiple inputs simultaneously so it is available to several users at the same time. Each upload has unique ID (timestamps in miliseconds) and is processed within its dedicated directory.

Third is providing results for further analysis. The Sleep Web App provides links to download the results. Users can do further statistical analysis using the tabular data stored in comma-separated values (CSV) format.

This Appendix A is organized as follows. Section A.2 describes how the Sleep Web App works. Next, the user manual is in Section A.3. The last Section A.4 describes the server-side installation process of the Sleep Web App.

## A.2 Sleep/Awake Classification in Sleep Web App

The Sleep Web App uses the sleep/awake classification method described in Chapter 5. It includes three steps. Figure A.1 is the block diagram.

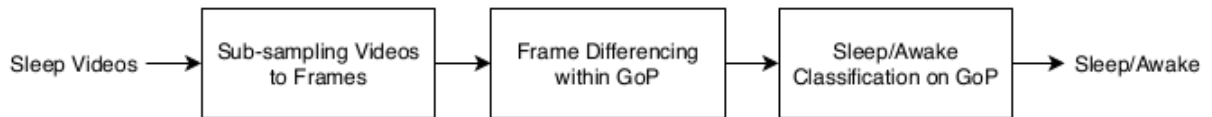


Fig. A.1. Block diagram for Sleep/Awake classification in Sleep Web App.

First, the sleep videos are converted into a sequence of frames at a constant subsampling rate, and the list of GoPs (Group of Pictures) are created from the frame sequences. The first step is implemented using Python with OpenCV (Open Source Computer Vision) Library [176].

Second, motion information is obtained through frame differencing within GoP as described in detail in Chapter 5. This step is implemented in C++ with OpenCV library.

Third, each GoP is classified as either sleep or awake by using deep learning model that was trained to learn sleep vs. awake patterns based on the motion index (described in detail in Chapter 5). The model used in Sleep Web App is LSTM-15 trained with weighed loss where the number of GoPs=15 had the best performance as shown in Table 5.3 in Chapter 5. This step is implemented in Python with TensorFlow [177] library.

Each steps requires specific environments to run. Without the Sleep Web App, supporting different user environments to run different sets of programming languages and libraries with specific versions would be extremely tedious work. By using a web

application, it is possible to provide the users with sleep/awake classification methods without asking them to install all the compilers and libraries that are used in the program.

By using a web-based application, it is possible to provide the sleep/awake classification to more researchers. Since the program deals with large data—sleep videos recorded all night, it requires lots of compute and processing power. Without web application, it is difficult to predict how much time it takes to run the program.

### **A.3 User Manual**

The Sleep Web Application provides Sleep/Awake labeling for pediatric sleep videos. To access the web app, please follow the link

<https://buddy-boy3.ecn.purdue.edu/~sleep/>.

This manual provides how to use the Sleep Web Application. The procedures in this manual are based on following conditions:

- One-night sleep videos recorded by Swann ADW-400 Digital Guardian Camera & Recorder
- Chrome browser
- Windows 7

#### **A.3.1 File Preparation**

This section explains how to prepare the video files to be uploaded to the server.

#### **Naming the Directories and Video Files**

The directories and files should be named in specific format. Figure A.2 is an example.

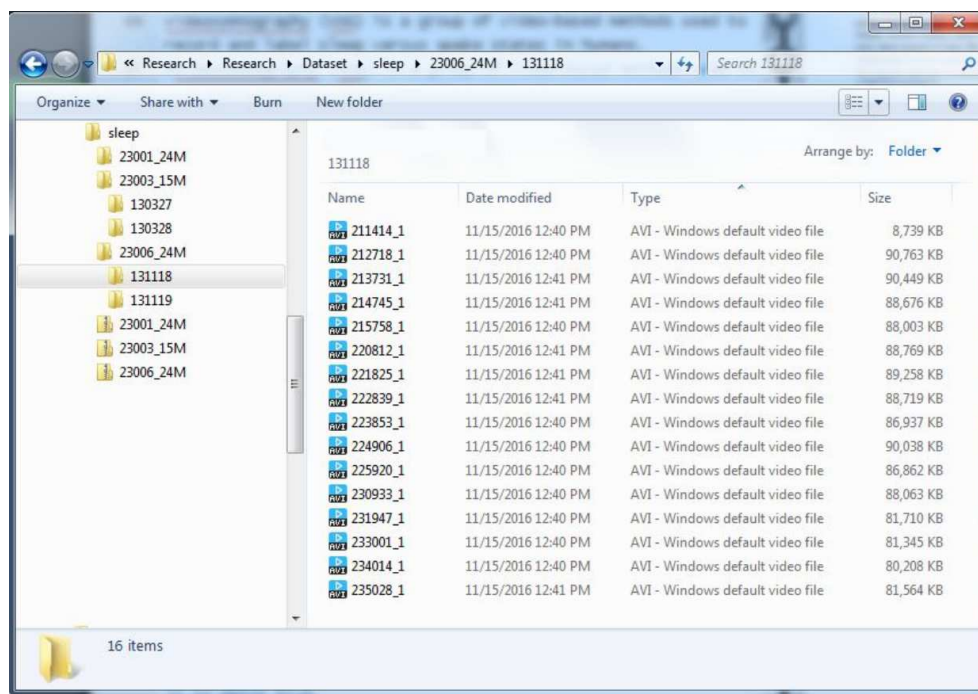


Fig. A.2. Sleep Web App Manual: File Compressing.

The top directory is the SubjectID\_Age\_Night (e.g. “23006\_24M\_1”). The subjectID is the 5-digit number, the Age is two-digit number in months and the Night is a number (any number of digits) for Night index. For SubjectID 23006, Age 24M, and the Night index 1, the directory name should be “23006\_24M\_1”

The second directory is the Date. There should be two Date directories for one-night sleep videos. The Date is 6-digit number, YYMMDD, where YY is year, MM is month, and DD is date. For one-night videos that were recorded from night in November 18, 2013 to the next morning in November 19, 2013, the two directories should be “131118” and “131119”.

Each Date directory contains sequence of AVI videos named with timestamps. The video filename is 6-digit timestamp followed by underscore and number 1, “\_1”. The 6-digit timestamp is HHMMSS, where HH is hour, MM is minute, and SS is second. A video that was recorded from 23 hrs 50 min 28 sec will have a filename of “235028\_1.AVI”. Videos recorded by Swann ADW-400 Digital Guardian Camera & Recorder automatically saves the videos with this filename pattern so there is no need to change the video names.

Here is the summary of how the directories and file names are structured:

```

/SubjectID_Age_Night
  /Date
    HHMMSS_1.AVI
    HHMMSS_1.AVI
  /Date
    HHMMSS_1.AVI
    HHMMSS_1.AVI

```

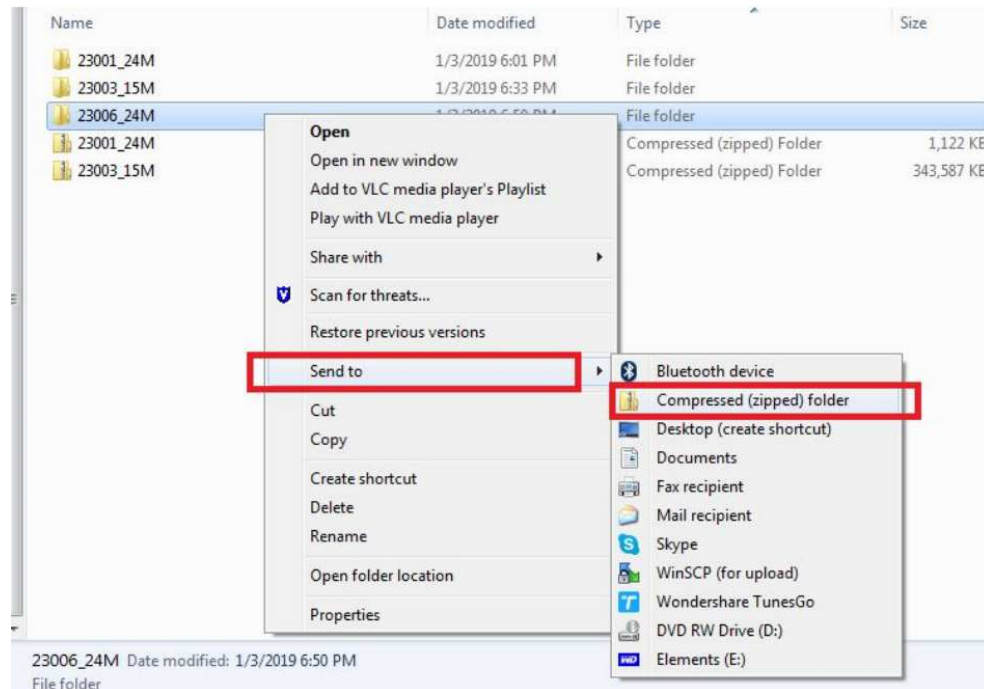


Fig. A.3. Sleep Web App Manual: File Compressing.

## File Compressing

The Sleep Web Application only accepts zip files. This section describes how to create zip file in Windows.

From the top directory named “SubjectID\_Age\_Night”, right click on the directory, select “Send to”, and select “Compressed (zipped) folder” as shown in Figure A.3

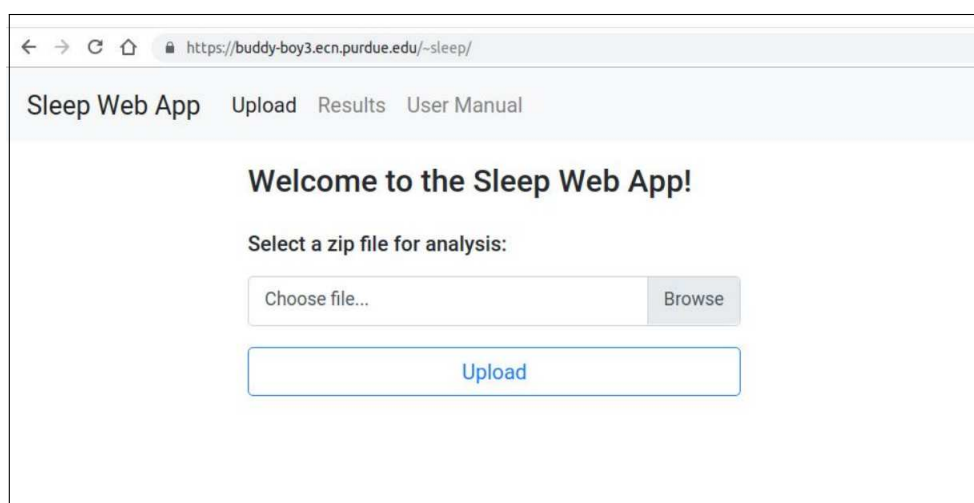
If it asks for what name to use for the “Compressed (zipped) Folder”, write the name of the top directory, SubjectID\_Age\_Night.

### A.3.2 File Uploading

The compressed (zipped) folder, basically the zip file, should be uploaded to the website, <https://buddy-boy3.ecn.purdue.edu/~sleep/>

Figure A.4 is an example of file uploading. First the user click on the “Browse” button, and then select the compressed (zipped) folder.





The screenshot shows a web browser window with the address bar displaying `https://buddy-boy3.ecn.purdue.edu/~sleep/`. The page has a navigation bar with links: **Sleep Web App**, [Upload](#), [Results](#), and [User Manual](#). The main content area features a heading **Welcome to the Sleep Web App!** followed by the instruction **Select a zip file for analysis:**. Below this is a file selection interface consisting of a text input field containing the placeholder text "Choose file..." and a "Browse" button. At the bottom of the form is a large "Upload" button.

Fig. A.4. Sleep Web App Manual: File Upload.

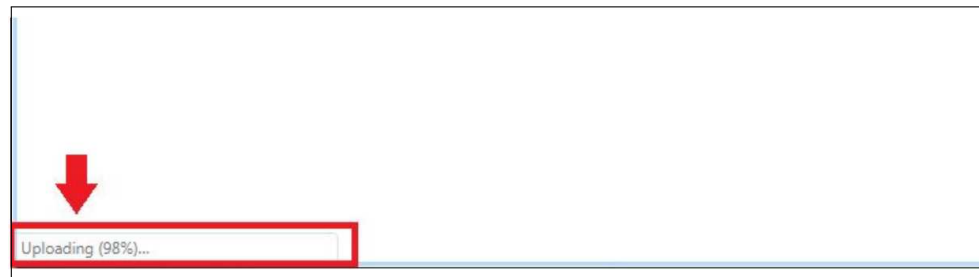


Fig. A.5. Sleep Web App Manual: File Upload.

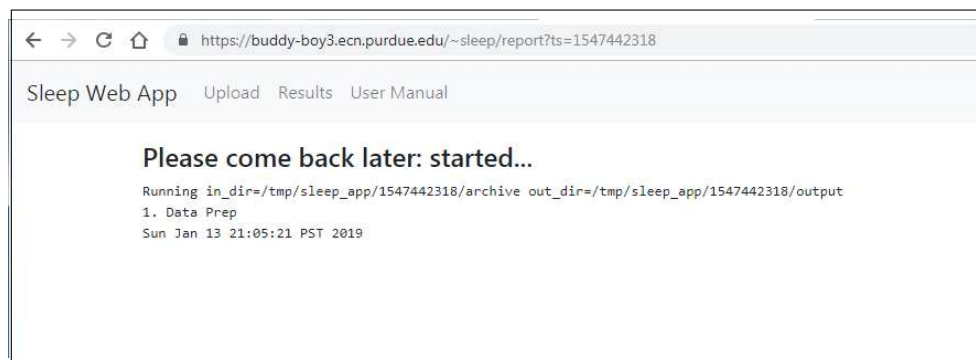


Fig. A.6. Sleep Web App Manual: File Upload.

Once the user click on the “Upload” button, the process begins. As shown in Figure A.5, the uploading progress is shown in the bottom-left corner of the browser.

Once the uploading is done, the video processing begins in the server and the page shown in Figure A.6 will show up.

Users can refresh the web browser to see the status of the processing or close the browser and check the results later on. Once the processing begins, it will run in the server regardless of whether the user closes the browser or not.

### A.3.3 Results

The results are listed on the “Results” menu as in Figure A.7.

The result for each upload can be viewed by clicking on the corresponding upload time. Figure A.8 is an example of the result.

← → ↺ 🏠 <https://buddy-boy3.ecn.purdue.edu/~sleep/results>

Sleep Web App   Upload   Results   User Manual

| Upload time (server time) | Status                         |
|---------------------------|--------------------------------|
| 2019-01-14 00:05:18       | success <a href="#">remove</a> |
| 2019-01-13 19:14:44       | success <a href="#">remove</a> |

Fig. A.7. Sleep Web App Manual: Results.

Sleep Web App

Upload

Results

User Manual

| ID    | Age | Night index | Sleep onset (HH:MM) | Sleep offset (HH:MM) | Sleep duration (min) | Awake duration (min) |
|-------|-----|-------------|---------------------|----------------------|----------------------|----------------------|
| 23003 | 15M | 5           | 23:39               | 00:20                | 42                   | 0                    |

Download summary csv file

Download results csv file

| ID    | Age | Night # | HH:MM | Sleep | Sleep ratio | Awake ratio |
|-------|-----|---------|-------|-------|-------------|-------------|
| 23003 | 15M | 5       | 23:39 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:40 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:41 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:42 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:43 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:44 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:45 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:46 | True  | 1.00        | 0.00        |
| 23003 | 15M | 5       | 23:47 | True  | 1.00        | 0.00        |

Fig. A.8. Sleep Web App Manual: Final result page. Download buttons for per-minute sleep analysis result and sleep summary results are provided.

## A.4 Environments

The Sleep Web App is implemented in one of the servers in Video and Image Processing Laboratory. The server uses Apache Web server [178] on Linux (Ubuntu 14.04.5 LTS). The Sleep Web App uses a python-based web development framework, Flask [179].

### A.4.1 Installations (system level)

This section describes how to deploy Anaconda-based web application to Apache Server.

Install Apache and Anaconda

---

```
sudo apt-get install apache2 apache2-bin apache2-dev
wget (url for Anaconda) & check md5sum
sudo bash Anaconda2-5.2.0-Linux-x86_64.sh -bfp /opt/anaconda2
```

---

Install conda packages and Web Server Gateway Interface (WSGI). WSGI enables python modules to be used in Apache server.

---

```
sudo su ## Login as superuser
export PATH=/opt/anaconda2/bin:\$PATH ## Add conda to your path
pip install mod_wsgi
mod_wsgi-express install-module ## check the outputs to this commands
    (used for Apache configuration in the next step)
conda install -c anaconda flask
sudo apt-get install libapache2-mod-wsgi python-dev
```

---

Three files, `wsgi.conf`, `wsgi.load`, and `000-default.conf`, need to be updated to update Apache Configurations. After making changes to each file, don't forget to restart Apache using the command `sudo service apache2 restart` to see if it reports any error.

First, open the file `/etc/apache2/mods-available/wsgi.conf` and add the following

---

```
<IfModule mod_wsgi.c>
    WSGIPythonHome /opt/anaconda2
    WSGIPythonPath /opt/anaconda2/lib/python2.7/site-packages
</IfModule>
```

---

Second, open the file `/etc/apache2/mods-available/wsgi.load` and add the following

---

```
LoadModule wsgi_module /usr/lib/apache2/modules/mod_wsgi-py27.so
```

---

Note: This is the output from ‘`mod_wsgi-express install-module`’ so yours could be different.

If the `LoadModule wsgi_module /usr/lib/apache2/modules/mod_wsgi.so` already exists in the file, comment it out. Otherwise, the apache server will run the default python instead of the python within Anaconda.

Enable the wsgi mod:

---

```
sudo a2enmod wsgi
```

---

It will output ‘`Module wsgi already enabled`’

Third, open the file `/etc/apache2/sites-available/000-default.conf` and add the following

---

```
WSGIDaemonProcess sleepapp python-home=/opt/anaconda2
    python-path=/var/www/flask/sleep
WSGIScriptAlias /~sleep /var/www/flask/sleep/sleepapp.wsgi
<Directory /var/www/flask/sleep>
    WSGIProcessGroup sleepapp
    WSGIApplicationGroup %{GLOBAL}
    WSGIScriptReloading On
    Order allow,deny
```

```

    Allow from all
    Require all granted
</Directory>

```

---

With the above configurations, the Apache server will run `sleepapp.wsgi` in `/var/www/flask/sleep`. In order to link this path in the root directory of an apache server, `/var/www`, with a directory in another location, `project_directory`, you can use symlink named `sleep` under `/var/www/flask/` and create symlink to the other directory

---

```

# ln -s [project_directory=/pub1/jeehyun/LSTM/sleep_website/server]
    [symlink_name=sleep]
ln -s /pub1/jeehyun/LSTM/sleep_website /var/www/flask/sleep

```

---

To have Apache server run the project, need to set the group ownership of both the symlink and the linked directory to `www-data`:

---

```

chown -h :www-data /var/www/flask/sleep
chown :www-data /var/www/flask/sleep/*

```

---

If the website is not loading, check followings to make sure all the settings are correct.

- See if Anaconda env is properly loading (e.g. python version, system path)
- the error logs stored in `/var/log/apache2/error.log`

#### A.4.2 Installations (for sleep/awake classification)

The sleep/awake classification is implemented in the same server as the Web server.

The libraries used for this method are TensorFlow [177] and OpenCV [176]. OpenCV is used with both C++ and Python 2.7. The libraries can be installed either in the server or in Anaconda environment.

## B. SOURCE CODE

The source code used in this thesis can be downloaded from Git repository of Video and Image Processing Laboratory (VIPER) or Purdue ECN server, *stargate.ecn.purdue.edu*.

The VHR-related source code is in the Blush project in VIPER Git repository:

---

```
https://lorenz.ecn.purdue.edu:3000/Blush
```

---

Source code related to Sleep studies are in the Sleep project in VIPER Git repository:

---

```
https://lorenz.ecn.purdue.edu:3000/sleep
```

---

Source code used in Chapter 6 can be found in Purdue ECN server, *stargate.ecn.purdue.edu*, in following path directory:

---

```
/home/stargate/a/sig/choe11/softwares/softwares_SSIAI2014
```

---