Improving Motion Robustness of Contact-less Monitoring of Heart Rate Using Video Analysis

Author: Pratik Sahindrakar

Supervisors:
Prof. Dr. Gerard de Haan (TU/e)
Dr. Ihor Kirenko (Philips Research)
Heart rate is the number of heartbeats per unit of time. Heart rate is one of the important physiological signals used by medical professionals to assist in the diagnosis and tracking of patients medical conditions. With every heartbeat a blood pressure pulse travels from the heart to the other body parts. Typically, it is referred as Blood Volume Pressure (BVP) or cardiac pulse or only pulse. A heart rate of a person can be detected if a cardiac pulse rate is detected at any other body part. Conventionally, a medical person measures cardiac pulse at any place that allows an artery to be compressed against a bone, such as at the neck, at the wrist, behind the knee, on the inside of the elbow, and near the ankle joint. Conventional automatic measurement of cardiac pulse include Electrocardiography (ECG), finger clip or earlobe pulse oxymetry sensors and chest strap. All above specified methods needs a direct contact with patients body to measure heart rate. Remote and non-contact measurements of the cardiac pulse can provide comfortable physiological assessment without clutter of wiring and electrodes. Photo-plethysmography (PPG) is a simple, non-invasive and low-cost optical technique that can be used to detect blood volume changes in the microvascular bed of tissue. Photo-plethysmographic signals are measured remotely (> 1m) using ambient light and a simple consumer level digital camera in a movie mode. Heart rate can be quantified up to several harmonics using PPG method. However, extracting the physiological signal from a video is affected by the motion of a subject. In literature, the algorithm developed by Division of health science and technology, Harvard-MIT claims to be the first motion improved algorithm to detect heart rate from a video analysis. In our study, we investigate above mentioned algorithm in detail and quantify its performance. The algorithm is verified for the motion robustness it claims using videos of a subject with different types of motion. It was found that the algorithm performs satisfactorily under stationary, slow translation and rotational motion with small rotation angle. Improvements in the existing motion tracking scheme used by the algorithm and signal processing of the algorithm are suggested, implemented and quantified. The improved motion tracking scheme involves skin pixels detection and patch tracking in successive frames of a video. To exploit further improvements, changes in a way signals are extracted from a video are suggested and verified for the performance improvement. It was found that a typical combination of red, green and blue traces gives better signal to noise ratio than the other. The reason why the particular combination works better than the others is also investigated from the light absorptivity curve of Oxyhemoglobin and De-oxyhemoglobin. Further improvement in the performance of the algorithm is achieved by extracting the differentiated signals from a video. After series of evolution in the existing algorithm, the final algorithm is more robust and computationally less expensive than the existing algorithm.
Acknowledgements

I would like to express my gratitude to my Philips supervisor, Dr. Ihor Kirenko, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas (e.g., algorithms, ethics, interaction with participants), and his assistance in writing reports (i.e., graduation report, technical note and technical paper), which have on occasion made me 'green' with envy. I would like to thank the other members working on the same project, Dr. Arno van Leest, Ingmar van Dijk and Vincent Jeanne for the assistance they provided at all levels of the research project.

A very special thanks goes out to Prof. Gerard de Haan, without whose motivation and encouragement I would not have considered a graduate career in Video and Image processing research. Prof. Haan is the one professor/teacher who truly made a difference in my life. It was under his tutelage that I developed a focus and became interested in video processing. He provided me with direction, technical support and became more of a mentor, than a professor. It was though his, persistence, understanding and kindness that I completed my graduate degree. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my father, without whose love, encouragement, I would not have journeyed so far. I would like to thank my house mates, who are also my best friends, for their patience, words of comfort and technical assistance. I express my gratitude towards my best friends, back at home, who have been invisible sources of motivation, encouragement and occasionally peace of mind. I must also acknowledge Sachin Bhardwaj for assisting in improving the technical report writing skills.

In conclusion, I recognize that this research would not have been possible without the assistance of Philips research, in particular the video and image processing group and Eindhoven University of Technology. I express my gratitude towards the group and the university.
Contents

1 Introduction..........................................................1
   1.1 Photo-plethysmography .....................................2
   1.2 Non-contact Photo-plethysmography .......................3
   1.3 Challenges with non-contact Photo-plethysmography ........4
   1.4 Existing attempt to improve motion robustness .............5

2 Blind Source Separation (BSS) algorithm ......................6
   2.1 Pulse measurement methodology ..........................7
   2.2 Heart rate extraction ......................................9
   2.3 Experiment set-up and recorded videos ...................9
   2.4 Quantifying the algorithm ................................10
   2.5 Analysis of the existing algorithm .......................11
      2.5.1 Improvement in Region of Interest selection and tracking ....11
      2.5.2 Improvement in pre-processing .......................12
      2.5.3 Improvement in post-processing ....................13
   2.6 Quantifying the improvements ...........................13
   2.7 Conclusion ................................................14

3 Improving the motion robustness .............................15
   3.1 Skin pixel detection ......................................15
   3.2 Motion tracking ..........................................17
   3.3 Selecting the motion estimation algorithm ..............17
   3.4 Pyramidal Lucas-Kanade feature tracker - LK method ....17
   3.5 Selecting the features to track ........................19
   3.6 Patch tracking ...........................................20
   3.7 BSS with patch tracking ................................21
   3.8 Results ..................................................22
   3.9 Conclusion ..............................................24

4 Improving the signal extraction from Region of Interest ....25
   4.1 HSI : Hue-saturation-intensity ..........................25
4.2 Results with HSI .................................................. 28
4.3 YCbCr ............................................................... 28
4.4 Results with YCbCr ................................................. 31
4.5 RGB ................................................................. 31
4.6 Results ............................................................... 37
4.7 Conclusion .......................................................... 37

5 Difference method ......................................................... 38
5.1 Average of the difference ........................................... 38
5.2 Results ............................................................... 39
5.3 Conclusion .......................................................... 40

6 Summary ................................................................. 41
6.1 Future work ......................................................... 42
6.2 Conclusion .......................................................... 44

A Independent component analysis (ICA) ............................... 48

B Quantifying the algorithm performance ............................... 49
Heart Rate (HR) is the number of heartbeats per unit of time, typically expressed as beats per minute (bpm). HR can vary as the body’s need to absorb oxygen and excrete carbon dioxide changes, such as during exercise or sleep. The measurement of HR is used by medical professionals to assist in the diagnosis and tracking of patients medical conditions. It is also used by individuals, such as athletes, who are interested in monitoring their HR to gain maximum efficiency from their training. A schematic representation of a healthy person’s Electrocardiogram (ECG) waveform is illustrated in Figure 1.1. The R wave to R wave interval (RR interval) in successive waveforms is the inverse of HR.
1.1 Photo-plethysmography

Joint. Physical phenomenon such as variation in air pressure, impedance, or strain can be used to detect cardio-vascular pulse. Currently, the gold standard techniques for measurement of a cardiac pulse such as the ECG require patients to wear adhesive gel patches (Fig. 1.2a) or a chest strap (Fig. 1.2d) that can cause skin irritation and discomfort. Commercial pulse oximetry sensors that attach to a fingertip or an earlobe (Fig. 1.2b,c) are also inconvenient for patients and spring-loaded clips can cause pain if worn for over a long period of time. The ability of monitoring patients physiological signal via a remote, non-contact means has a promise for improving access and enhancing the delivery of primary healthcare and hence it received a lot of attention in recent time. For example, the idea of performing physiological measurement on a face was first postulated by Pavlidis and associates [22] and later demonstrated by analysis of facial thermal videos [12, 14].

![Figure 1.2: Presently used standard techniques for measurement of the cardiac pulse. (a) ECG with electrodes connected using adhesive gel patches (b) Pulse oximetry sensor attached to an earlobe or (c) Pulse oximetry sensor attached to a fingertip (d) Chest strap.](image)

Although non-contact methods may not be able to provide details concerning cardiac electrical conduction that ECG offers, these methods can now enable long-term monitoring of other physiological signals such as HR or respiratory rate by acquiring them continuously in an unobtrusive and comfortable manner. Beyond that, such a technology would also minimise the amount of cabling and clutter associated with neonatal ICU monitoring, long-term epilepsy monitoring, burn or trauma patient monitoring, sleep studies, and other cases where a continuous measure of HR is important.

1.1 Photo-plethysmography

Photo-plethysmography (PPG) is a simple, non-invasive and low-cost optical technique that can be used to detect blood volume changes in the microvascular bed of tissue (introduced in 1930’s [12]). The basic form of PPG technology requires only few opto-electronic components: a light source to illuminate a tissue (e.g. skin), and a photo-detector to measure the small variations in light intensity associated with changes in perfusion in the catchment volume. A general set-up for capturing the PPG signal is as shown in Figure 1.3.

The interaction of light with a biological tissue is complex and includes the optical processes of (multiple) scattering, absorption, reflection, transmission and fluorescence. Dominantly light is absorbed by bones, tissues and blood. Moreover, blood absorbs more light than the surrounding tissue. The key factors that can affect the amount of light absorbed by the blood are: the blood
volume, blood vessel wall movement and the orientation of red blood cells (RBC) [1]. The skin, being the easiest tissue available, is popular for PPG analysis. The skin is divided in three layers viz. Epidermis - the visible layer of skin, Dermis - the layer of skin below Epidermis and Subcutaneous (also called Hypodermis) - the layer below Dermis (Figure 1.4a). The skin is richly perfused and has the congregation of multiple veins called as venous plexus in Dermis and Subcutaneous. With each cardiac cycle heart pumps blood to the periphery. Even though the pressure pulse is somewhat damped by the time it reaches the skin, it is enough to distend the arteries and arterioles in the subcutaneous tissue. It is relatively easy to detect the pulsatile component of the cardiac cycle in the skin. A PPG is often obtained by using a pulse-oximeter sensor. If a pulse-oximeter is attached without compressing the skin, the changes in blood volume caused by the cardio-vascular pressure pulse can also be detected by illuminating the skin with a light from a light-emitting diode (LED) and then measuring the amount of light either transmitted or reflected to a photodiode (Figure 1.3). A typical distribution of the light absorption is illustrated in Figure 1.4b [15]. The DC component of the signal is attributable to the bulk absorption of the light by the skin tissue, while the AC component is directly attributable to the variation in the blood volume in the skin caused by the cardio-vascular pressure pulse. A conventional pulse oximeter monitors the perfusion of blood to the dermis and subcutaneous tissue of the skin (Figure 1.4a). Each cardiac cycle appears as a peak on an extracted PPG signal, as seen in the figure 1.5. As blood flow to the skin can be modulated by multiple other physiological systems, the PPG can also be used to monitor breathing, hypovolemia (a state of decreased blood volume), and other circulatory conditions.

Conventionally, PPG has been performed with a dedicated light source, typically red and/or infra-red (IR) wavelengths. Due to the historical emphasis of PPG on pulse oxymetry and the associated need to sample relatively deep (e.g., 1 mm) veins and arteries, the visible spectrum (with a shallower penetration depth in the skin) has often been ignored as a light source for PPG. The publications describing non-red visible light sources for PPG (e.g. green) are either recent [13, 18] or relatively old [9, 20], and, in all cases, contact probes were used. The ambient visible light is often considered a source of noise [7, 26] when using IR light sources and detectors sensitive for IR and visible light.

1.2 Non-contact Photo-plethysmography

Remote, non-contact pulse oxymetry and PPG imaging have been explored only relatively recently [16, 29] which uses SpO(2) camera technology and three different wavelengths LED. The recent work [28] has shown that pulse measurements can be acquired using digital camcorders with normal ambient light as an illumination source. Typically, the G channel, out of RGB channels, contains the strongest
1.3 Challenges with non-contact Photo-plethysmography

If the PPG is obtained by a dedicated source of light and an unmoving skin patch then the PPG signal obtained is very clean, similar to the signal in Figure 1.5. The challenge is to get a good PPG signal from a moving skin area. PPG is known to be susceptible to the motion-induced signal corruption. Overcoming motion artefacts presents one of the most challenging problems. The PPG signal is corrupted due to the changes in illumination of the skin area. To better understand this we refer to the Figure 1.6 which shows a typical scenario in the non-contact PPG measurement. A light source (ambient light or dedicated source) illuminates the skin area. Some part of the light, called specular reflection, is reflected by the skin. Some part of the light penetrates through the skin and...
gets diffused back with its complex interaction with vasculature in the skin layers. The \textit{diffused light} carries a plethysmographic signal. For simplicity, in our model we are not considering the part of a light which penetrates through the skin but get diffused before reaching the vasculature in the skin. Such a diffused light would not carry any plethysmographic signal. For further references diffused light is the one which carries plethysmographic signals. The signal received by a photodetector is a combination of a specular reflection and a diffused light. If the skin area under observation starts moving then due to the uneven nature of a skin, light intensity received by a photo-detector changes as the specular reflection and the diffused light intensity changes.

![Diagram of light interaction with skin](image)

Figure 1.6: The interaction of a light with the skin is illustrated. In general, the amount of light received by a photodetector will change if the angle of a skin patch with a source and a receiver is changed.

Small and slow motion in a skin area causes the specular reflection to change slowly. A resultant PPG signal superimposed on a slow varying large intensity component is detectable. On the other hand, a large and rapid motion of the skin area causes a fast variation in the specular reflection component. A resultant small PPG signal superimposed on a large intensity variations is unrecognisable. In order to develop a clinically useful technology, there is a need for an ancillary functionality such as motion artefact reduction through efficient and robust image analysis.

### 1.4 Existing attempt to improve motion robustness

In the literature, the algorithms developed by Division of health science and technology, Harvard-MIT [24] claims to be the first attempt for low-cost accurate video-based method for contact free HR measurement that is automated and motion tolerant. The algorithms is based on a relatively new mathematical technique called \textit{Blind Source Separation (BSS)}. Considering that this is the first attempt of a video-based method for a contact-less HR measurement and the good results claimed by this algorithm, we decided to study the algorithm in detail and verify the results it claimed.

In this chapter we saw the basic theory of Photoplethysmography, its application for HR detection and the challenges it poses with the motion. In Chapter 2, we will see the existing motion robust algorithm of HR detection from video analysis introduced in the literature, suggest and implement some basic improvements in the same algorithm. In Chapter 3, improvements in the motion robustness are investigated. In Chapter 4, robust signal extraction strategies are investigated. Further improvement in signal extraction in the form of Difference method is suggested in Chapter 5. An overview of the findings of the study is summarised in Chapter 6.
Blind Source Separation (BSS) algorithm

In this chapter first we will introduce the algorithm developed by Division of health science and technology, Harvard-MIT [24]. Before going into specific details of the algorithm, we will see an overview of the HR extraction from video. A generalised flowchart of the HR extraction algorithm from a colour video is illustrated in Figure 2.1. The input system (block 1) reads video frames either from a camera or from a colour video of a subject stored in a storage device. In each frame Region of Interest (ROI)(s) is defined and it is tracked in successive frames to avoid abrupt changes in the extracted signal (block 2). From the selected ROI in each frame, a signal is extracted according to the algorithm in block 3. For a physiological signal extraction, a signal over some constant time period is required such that it covers a sufficient amount of information which can be extracted by further signal processing. It is assured that the signal of sufficient time length is available before the signal processing can continue (block 4). In pre-processing (block 5) available signal is refined to eliminate unwanted noise (errors in tracking, motion of a subject etc.). HR signal is then extracted from a refined signal using frequency spectrum analysis(block 6). To decrease the possibility of HR mis-prediction due to the noise (insufficient refinement of a signal, motion of a subject etc.), post-processing (block 7) corrects the Estimated Heart Rate (EHR) by considering the history. The corrected EHR is then recorded or displayed (block 8).

In our study of HR extraction from video analysis algorithms, we refer to the algorithms depending on the ROI selection and tracking method and the way of signal processing. We refer to the implementation of the HR extraction algorithm based on face detection and BSS introduced in [24] as FD-BSS. The FD-BSS provides remote comfortable measurement of the cardiac pulse without an electrode. It extracts the BVP signal from a colour video recording of a human face. It is based on automatic face tracking along with the BSS of the colour channels into independent components. BSS is a technique for a noise removal from physiological signals. BSS refers to the separation of a set of signals from a set of mixed signals, without the aid of an information (or with very little information) about the source signals or the mixing process. Typically, the observations are acquired from the output of a set of sensors, where each sensor receives a different combination of the source signals. There are several methods of BSS but FD-BSS focuses on BSS by Independent Component Analysis (ICA) [8]. ICA is a technique for uncovering the independent source signals from a set of observations that are composed of linear mixtures of the underlying sources. ICA finds the independent components (factors, latent variables or sources) by maximising the statistical independence of the estimated components. The use of this fairly new technique in biomedical signal analysis is rapidly expanding, e.g. in noise removal from electrocardiogram (ECG) and electroencephalogram (EEG) recordings, separation
2.1 Pulse measurement methodology

Figure 2.1: A generalised flowchart for a HR extraction algorithm from video analysis is illustrated. A mapping of steps of the FD-BSS on the generalised flowchart is illustrated by the projected dotted lines.

of fetal and maternal ECGs recorded simultaneously, as well as detection of event related regions of activity in functional magnetic resonance imaging (fMRI) experiments. ICA has also been applied to reduce motion artefacts in PPG measurements [19, 30]. Readers can refer to the short summary of ICA in Appendix A. For improving the performance of an algorithm improvements in different blocks of the flowchart (Figure 2.1) are necessary. On the right side of the flowchart, mapping of the steps of FD-BSS on the generalised flowchart is illustrated. In the next section, we discuss FD-BSS in details.

2.1 Pulse measurement methodology

An overview of the general step in signal extraction from a video in FD-BSS is illustrated in Figure 2.2. First, an automated face tracker is used to detect face within a video frames and locating the ROI for each video frame [Fig. 2.2B]. The Viola-Jones face detection algorithm implemented in Open Computer Vision (OpenCV) library is used to obtain the coordinates of the face location [5]. A pre-trained frontal face classifier available with OpenCV 2.0 is used. For each face detected, the algorithm returns x- and y-coordinates along with the height and width that define a box around face of a subject. In Figure 2.2b the outer box shows the face detected by the Viola-Jones method. From this box, the centre 60% width and full height is selected as the ROI for subsequent processing. The ROI is marked by the inner box. To prevent face segmentation errors from affecting the performance of the algorithm, the face coordinates from a previous frame were used if no faces were detected. Also, if multiple faces are detected then the face box which has the top-left coordinate closer to the top-left coordinate of a face box in a previous frame is selected.

The ROI was then separated into the RGB sub-images [Fig. 2.2c] and spatially averaged over all pixels in the ROI to yield the average red, blue and green values for each frame to form raw R,G and B traces which we call $x_1(t), x_2(t),$ and $x_3(t)$ respectively [Fig. 2.2d]. The program for extracting raw trace from a video is implemented in C++. The raw traces over 5 sec of a sample video are illustrated in Figure 2.3a.

The program for processing the raw traces is implemented in Matlab. In this program, first the
2.1 Pulse measurement methodology

Figure 2.2: Above figure illustrates the process of extraction of raw RGB trace from a video sequence. 
(a) A video sequence is illustrated along the time axis. (b) A frame at the time instant \( t_3 \) is illustrated along with the ROI marked with solid line box. (c) The ROI is separated in R, G and B sub-images. Average taken over R,G and B components of ROI are placed in traces as shown by curved arrows going from sub-image (c) to sub-image (d). (d) Shows the raw RGB traces.

(a) Figure illustrates the raw R,G,B traces obtained by the process explained in Figure 2.2 for a sample video sequence. As variations due to BVP on R,G and B traces are very small, the wave nature of these traces is not clearly visible.

(b) Figure illustrates the shifting of Window with \( \text{WindowOverlap} \) along the time axis. The dashed lines represents the time instances at which HR signal is calculated.

Figure 2.3: Figure 2.3a illustrates the raw traces obtained from a sample video. Figure 2.3b illustrates the shifting Window used for in signal processing.
raw RGB traces are normalised as follows:

\[ x_\prime_i(t) = \frac{x_i(t) - \mu_i}{\sigma_i} \]  

(2.1)

for each \( i=1,2,3 \) where \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation of \( x_i(t) \) respectively. The normalisation transforms \( x_i(t) \) to \( x_\prime_i(t) \) which has zero-mean and unit variance.

### 2.2 Heart rate extraction

In FD-BSS subsequent processing is performed using a 30sec moving window with 96.7% overlap (1sec increment). As shown in Figure 2.3b, Window represents the part of the traces which are considered for extracting the HR at the given instant. A Window moves along the time axis with increment of WindowOverlap. Normalisation of the raw traces in a Window is performed using the formula in Equation 2.1. The normalised raw traces are then decomposed into three independent source signals using ICA. In FD-BSS, ICA by the Joint Approximate Diagonalization of Eigenmatrices (JADE) algorithm developed by Cardoso [4,6] is used.

The separated source signal was smoothed using a five-point moving average filter and bandpass filter (128-point Hamming window, 0.7 to 4 Hz). Although there is no ordering of the ICA components, FD-BSS selects the second component for further analysis of HR as it typically contained a strong plethysmographic signal. Finally, Fast Fourier Transform (FFT) of the selected source signal (i.e. second component) is taken to get the power spectrum. The pulse frequency is designated as the frequency that corresponded to the highest power of the spectrum within an operational frequency band. According to FD-BSS, the operational range is set to \([0.7, 4]\) Hz (corresponding to \([42, 240]\) bpm). The signal processing, described above, is the pre-processing of a signal.

Despite the application of ICA in FD-BSS, the pulse frequency computation may occasionally be affected by a noise. To address this issue, FD-BSS utilise the historical estimations of the pulse frequency to reject artefacts by fixing a threshold for maximum change in pulse rate between successive measurements (taken 1 s apart). If the difference between the current pulse rate estimation and the last computed value exceeded the threshold (FD-BSS used a threshold of 12 bpm in their experiments), the algorithm rejects it and searches the operational frequency range for the frequency corresponding to the next highest power that meets this constraint. If no frequency peaks that met the criteria were located, then FD-BSS retained the last computed pulse frequency estimation.

### 2.3 Experiment set-up and recorded videos

For validating the motion robustness of FD-BSS, various test videos are recorded. For recording these videos uEye video camera, which captures video at 20fps, 3channel (red, green, blue), 8 bit/channel and VGA resolution, is used. The set-up used for capturing videos is illustrated in Figure 2.4. The distance between a subject and the camera, when the videos are recorded, is approximately 1.5m. A subject in a video has European fair skin tone. The videos are recorded in a lab room and they are classified in three groups viz. color, translation and rotation. In group color, there are two videos with no motion. The color01.rgb is recorded using TL (fluorescent) lights and color02.rgb is recorded using only daylight. The videos in rest of the two groups are recorded in white light only. The videos in group translation and rotation has translational and rotational motion respectively. The videos are stored in PFSPD (Philips File Standard for Pictoral Data) file format on an external hard-disk. A physiological monitoring device, the Nexus 10, is connected via Bluetooth to a computer(Figure 2.4c and 2.4d). Using Nexus 10 device ECG signal (time waveform and HR data) is stored on the external...
A quantitative comparison of the performance is essential to keep track of the development of an algorithm. In the HR extraction algorithm, every time the Window is shifted by WindowOverlap we get a power spectrum and EHR. Stacking the power spectrums together over the time gives a spectrogram while stacking EHR gives an Estimated Heart Rate Trace (EHRT). The performance of an algorithm depends on how clean the spectrogram is and how close the EHRT is to the RHRT at any time instant. For quantifying the performance of an algorithm, two parameters are used viz. Signal to Noise Ratio (SNR) and Root Mean Square Error (RMSE). The SNR of a spectrogram indicates the cleanliness of the spectrogram. Spectrogram with higher SNR are better and they give better heart rate estimates. The RMSE value represents the accuracy of an EHRT. EHRT with lower RMSE values means that the HR estimated by video analysis is closer to the HR measured by the standard ECG method. The method by which SNR and RMSE values from spectrogram and EHRT are obtained is described in Appendix B. As SNR is a ratio of energy, it does not have any unit. The RMSE value does not have any unit as well. For a HR estimation algorithm SNR should be as high as possible and RMSE should be as low as possible. As the SNR of a spectrogram is nothing but the average of SNR of all the power spectrums in the spectrogram, it is very difficult to define a threshold for defining clean spectrogram. From observation of several spectrograms and their corresponding SNR values, a spectrogram having SNR above 0.4 can be considered sufficiently clean. Similarly, a RMSE value below 5 can be considered as a good result for an accuracy of the algorithm.
2.5 Analysis of the existing algorithm

Figure 2.5: Figure illustrates the degree of rotational motion in four recorded videos in the group rotation. A line through tip of a nose and the camera is considered as a reference for measuring angles. A dashed line at 60° on both sides of the reference line is drawn. A red cone on each side of reference shows the range of maximum rotation of a subject. Starting from the reference line if a person rotates head, red cone is the area in which person starts rotating in opposite direction. The rotation00, rotation01 videos are recorded in an ambient light and maximum rotation in the range 65°-75° and 50°-60° respectively. The rotation02, rotation03 videos are recorded using a light source on one side of a subject’s face such that other half of a face has shadows. The maximum degree of rotation in case of rotation02 and 03 is 80°-90° and 45°-55° respectively.

For quantifying FD-BSS in [24] only RMSE is used which does not provide any information about the cleanliness of a spectrogram. In general it can be concluded that SNR gives performance measure for pre-processing of a signal. The RMSE value gives performance measure for pre-processing, heart rate extraction strategy from frequency analysis and post-processing.

The post-processing can eliminate the errors due to substandard performance of pre-processing thus improving i.e. decreasing RMSE.

2.5 Analysis of the existing algorithm

To evaluate the performance of FD-BSS algorithm, it has been tested on the set of videos. The SNR and RMSE for various test videos are illustrated in Figure 2.8. In Figure 2.8b, a poor performance of FD-BSS for rotation00 and 02 can be noticed. In some cases (translation 00 and 01) though SNR value is poor, RMSE value is considerably better. The improvement in RMSE in such cases is a result of the post-processing. Next, we discuss the causes of poor performance of FD-BSS and possible solutions to overcome those problems.

2.5.1 Improvement in Region of Interest selection and tracking

In FD-BSS, the ROI is selected as the middle 60% width and full height of the face box. The big ROI selected this way has advantages as well disadvantages. The advantages include: less complexity in tracking, improvement in SNR due to average over large number of pixels and simple calculations. However, the big ROI selected by FD-BSS includes many non-skin pixels which do not carry plethysmographic information and corrupts the raw RGB traces. Moreover, error in face detection causes selection of a region without any skin pixels which causes occasional spikes in the raw RGB traces. In [24] FD-BSS is tested for slow movements of subjects such as tilting the head sideways, looking
2.5 Analysis of the existing algorithm

In above figure, blue and red curves represents frequency response for smoothing filter and band pass filter used by FD-BSS.

up/down, leaning forward/backward and talking. We have tested FD-BSS with various degree of movements to verify its rigidity. It was observed that with fast movements the number of face mis-predictions increases. A lot of mis-predictions are noticed in rotational movement. The frontal face classifier used for face detection works satisfactorily until face is roughly within $\pm 60^\circ$ (calculated empirically) from the camera to nose-tip reference line (Figure 2.5 illustrates this angle using dotted line). Beyond this angle frontal face classifier fails to detect a face correctly. Due to lot of mis-predictions, a raw data obtained for ICA analysis is corrupted which results in noisy spectrogram and wrong EHR. This explains the poor performance of FD-BSS for rotation00 and rotation02. To improve the motion robustness, in case of rotational motion, not only frontal face classifier but also side face classifier would be needed. Method for improving ROI selection and tracking are investigated in Chapter 3.

2.5.2 Improvement in pre-processing

In FD-BSS, the selected independent component, after the ICA, is smoothed and band pass filtered. The frequency response for five point smoothing filter and 128 point hamming band pass filter, as used by the pre-processing of the raw signals, is illustrated in Fig. 2.6. The heart rate range of FD-BSS is $[42, 240]$ bpm (Sec. 2.2). The five point averaging filter suppresses the higher frequencies in the same range violating the defined heart rate range. The use of averaging filter suppresses the higher frequency noise, thus helps improve SNR. A tread-off between SNR and HR range can be possible. In any case, either the claim of FD-BSS of detecting HR in the range $[42, 240]$ bpm or the use of averaging filter is questionable. As we prefer the extended HR range over SNR, we do not use averaging filter. Removing the averaging filter results in decrease of SNR.

Secondly, we saw that FD-BSS selects second component out of the three obtained independent components for extracting HR information. Even though there is no ordering of the ICA components, FD-BSS does it for the sake of simplicity and automation. This works because in most of the cases strong plethysmographic signal is seen in second component. Also the post-processing of FD-BSS corrects an error due to mis-prediction in case second component does not have plethysmographic signal. Though post-processing improves RMSE, it can not improve the poor SNR. If select the independent component which has a highest peak, improvements in SNR and RMSE have been observed.
2.6 Quantifying the improvements

Figure 2.7: Above figure gives an example of the problem caused by the to post-processing used in FD-BSS. The figure shows three HR traces for rotation02 video sequence. The RHRT obtained from Nexus 10 device, the trace from FD-BSS and the trace from FD-BSS-WoPP on rotation02 is illustrated. The dashed lines besides RHRT represents RHRT\(\pm5\)bpm reference lines. The traces from FD-BSS and FD-BSS-WoPP are noisy, a cumulative effect of poor face tracking due to the large motion in rotation02 (Figure 2.5) and poor separation of independent components by ICA. The trace from FD-BSS is trapped to \(\pm12\)bpm across first wrong prediction which is 42. The trace from FD-BSS-WoPP however, gives HR close to the reference for some estimates. This shows that there were some correct estimates which were discarded by the thresholding applied in post-processing.

2.5.3 Improvement in post-processing

The implementation of FD-BSS [24] takes advantage of historical estimation of HR to improve RMSE in case EHR is out of threshold. If a peak is not found within the range of \(\pm12\) bpm across the previous estimate then the previous estimate is selected as the current EHR. However, the value to be used as a previous estimate for the first estimation is not specified. Assuming that the first estimate does not have method to obtain the previous estimate, the first estimate is directly selected as the EHR. This has worked in many cases when first estimate is correct. In our test cases it worked for all videos except for rotation02. The problem caused by post-processing is illustrated in Figure 2.7. This example shows that the post-processing needs more sophistication to not get trapped due to thresholding. We could think of improving it by not just considering the historical estimate but also the SNR. In our implementation we do not use any post-processing.

2.6 Quantifying the improvements

The improvements in pre and post-processing (block 5 and 7 in Figure 2.1) of FD-BSS gives us Face Detection-Blind Source Separation with Improved Pre and Post Processing (FD-BSS-IPPP). To compare the performance of FD-BSS and FD-BSS-IPPP we test both the algorithms on our set of test videos. As in FD-BSS-IPPP we do not apply any post-processing we compare the results of FD-BSS-IPPP with the results obtained without post-processing for FD-BSS for fair comparison. Let us call this algorithm as FD-BSS-WoPP. As specified in Section 2.4 we compare these algorithms for quality of the spectrogram (SNR) and accuracy of EHR. In Figure 2.8 results are illustrated in a bar graph of SNR and RMSE values. In Figure 2.8a, it can be observed that SNR for FD-BSS-IPPP has decreased. In FD-BSS-IPPP, smoothing filter is not used in the pre-processing. The noise in high
2.7 Conclusion

In this chapter, we saw the implementation details of FD-BSS which is claimed to be the first motion improved non-contact HR estimation algorithm from video analysis. The motion robustness of this algorithm is evaluated using various test videos in which subject is either stationary, in translation motion or in rotational motion. We suggested and implemented three important improvements in FD-BSS to improve its performance. In the last section, suggested improvement in pre and post-processing (block 5 and 7 of the flowchart in Figure 2.1) of FD-BSS are implemented to give FD-BSS-IPPP. Though the improvements in FD-BSS-IPPP are not significant, it eliminates the potential problems we discussed, caused by the pre- and post-processing. It is expected that the improvements in the ROI selection and tracking can improve the performance of FD-BSS-IPPP.

frequency components causes SNR to decreases. In Figure 2.8b, for some videos, FD-BSS-IPPP has improved RMSE value. However, in most of the cases because of high frequency noises and lack of post-processing it has produced very poor results.

Considering the RMSE values for color01, color02, translation02 and rotation03 videos (in which person is either stationary, has slow translation motion or rotational motion in small angle) we can conclude that FD-BSS-IPPP has shown some improvement over FD-BSS. The performance for other videos can be improved with better ROI selection and tracking techniques. In next chapter, the problem of ROI selection and tracking i.e. improvement on block 2 of Figure 2.1 is discussed.

2.7 Conclusion

Figure 2.8: Performance comparison of FD-BSS, FD-BSS-WoPP and FD-BSS-IPPP.
Improving the motion robustness

At the end of the previous chapter we have introduced FD-BSS-IPPP which improves pre and post-processing of FD-BSS. In this chapter, we deal with the first and more demanding problem of ROI selecting and tracking. Considering the problems in ROI selection and tracking (Section 2.5.1) we come to some requirements for improving HR analysis in presence of motion which are as follows.

- We must be able to track the ROI under consideration in successive frames of a video under any type of motion.
- The ROI must consist of skin pixels only.

In FD-BSS, the problem of ROI selection and tracking is approached by simple face detection algorithm. The soul of selecting the middle 60% width of the face box is to restrict the ROI to the skin pixels. To achieve the task of selecting the skin pixels, a better approach is to do skin pixels detection. Also, face detection has various shortcomings which hampers the performance of HR extraction due to corruption of the raw RGB traces. For better results, the skin pixels must be tracked in successive frames. To handle the problem of ROI selection and tracking on face, we propose a better approach. Instead of having a big box bounding middle part of a face, which also includes non-skin pixels, we propose to have many small boxes on a face consisting of skin pixels only. These small square boxes must be tracked in successive video frames. This idea leaves us with the following tasks:

- Selecting the points for tracking in a frame.
- Discarding the points which are not in skin pixels region.
- Tracking the remaining points in successive frames.

The question of selecting the points for tracking becomes intuitive once the mathematical ground for motion tracking is led out. Hence, in following section we first go through skin pixels detection, selection of the motion tracking algorithm and selection of the points to track.

3.1 Skin pixel detection

In recent years, there has been a growing interest in the problem of skin segmentation, which aims to detect skin region in an image. Knowledge on skin colour is often used for locating skin-coloured pixels
because of its advantages: low computation cost and robustness against geometrical transformations. In literature, several algorithms have been proposed for skin colour pixel classification. They include piecewise linear classifiers, the Bayesian classifiers with the histogram technique, Guassian classifiers, and the multilayer perceptron. A systematic study and comparison of these algorithms is given by Phung et al. [23]. Videos recorded under natural environments are frequently subjected to illumination variations which affect their colour appearance. The study of skin pixel detection under changing illumination on existing skin-pixel detection algorithm is performed by Marinkauppi et al. [21].

As the skin-pixel detection is not the main focus of our study, a simple, real time skin-pixel detection algorithm using OpenCV functions is implemented. An overview of the algorithm is illustrated in Figure 3.1. For the first frame, a ROI (a box) bounding the face of a subject is specified. For skin detection, this ROI is extracted from the frame and converted into HSV colour space. The HSV ROI image (\textit{imgHsvROI}) goes under the calculation to give an initial skin mask and a hue image (\textit{imgHueROI}). The initial skin mask is calculate by a simple piecewise HSV linear classifiers. All pixels in the following HSV range are marked as the skin pixels: \(H = 0..180, S = 96..255, V = 21..255\) (calculated empirically). If the frame under consideration is the first frame of a video then a 16 point hue histogram from the middle 60% of the \textit{imgHueROI} (i.e. \textit{imgHist}) is calculated. Next, for each frame a probability map using \textit{imgHsvROI} and hue histogram is calculated. The probability map indicates the probability of a pixel being a skin pixel scaled in 0 to 255. The \textit{cvCalcBackProject} function in OpenCV computes the probability map (\textit{imgBackProject}) given the histogram and \textit{imgHueROI}. A bit-wise ANDing (\textit{cvAnd}) operation on an initial estimate of a skin mask and \textit{imgBackProject} gives a refined skin mask. The skin mask is further processed (threshold, erosion and dilation operations) to give the final skin mask in \textit{imgMask}. In this algorithm, the initial ROI is tracked using \textit{cvCamshift} (not shown in Figure 3.1 for clarity) as it is needed for skin mask calculation of future frames.

To improve the performance under illumination changes, the histogram can be updated per frame or after some defined number of frames. This increases the complexity of the algorithm. In case of our test videos, there are no videos which have large illumination or colour changes. Thus, calculating hue histogram for the first frame works satisfactorily. Depending on the requirement of the system, skin-pixel detection can be improved or replaced by a better algorithm in the future.
Using skin pixel detection, as specified in Figure 3.1, as a replacement to the face detection in FD-BSS-IPPP, we implement an algorithm referred as Skin Map - Blind source separation with Improved Pre and Post Processing (SM-BSS-IPPP). The results of test on the videos in our test set are illustrated in Section 3.8.

3.2 Motion tracking

Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may be represented by a translational model or many other models that can approximate the motion of a real video camera, such as rotation and translation in all three dimensions and zoom.

Motion estimation is a powerful means to improve a variety of video processing systems. Over the years, motion estimation algorithms have been developed. Initially, algorithms repeated their calculations for every pixel in the image sequence. Later, they were repeated on a block-by-block basis, where blocks are groups of pixels. Most recently, researchers aim at repeating calculations per object, where objects are often defined as the group of pixels exhibiting a similar type of motion. Motion estimation algorithms are classified as pixel based, block based and object based algorithms. Various algorithms are introduced in [10].

3.3 Selecting the motion estimation algorithm

The various algorithms presented in [10] have their own advantages and disadvantages. Over the time these algorithms have developed and became computationally efficient. The 3DRS algorithm [11] is the most computationally efficient algorithm for calculating motion vector for each pixel or box in a frame. However, if we need to calculate motion vector for a particular point in successive frames, motion vectors for each pixel in a frame needs to be calculated as the candidate vectors for calculating motion vector in the successive frame are selected from the 3-D neighbourhood (spatial and temporal). On the other hand, Pyramidal Lucas-Kanade feature tracker algorithm (Pyramidal LK method) focuses on calculating motion vector for given points, called features [2]. Moreover, we specified our requirement as to follow various points in the skin pixel region in successive frames. Thus, for our need Pyramidal LK algorithm is suitable for motion tracking. Apart from that, an efficient implementation of Pyramidal LK method is available in OpenCV library in the function cvCalcOpticalFlowPyrLK() which makes it easier to access with other OpenCV functions [3]. Next, we briefly go through the Pyramidal LK method implemented in OpenCV which is often used in the context of motion estimation.

3.4 Pyramidal Lucas-Kanade feature tracker - LK method

The Lucas-Kanade (LK) algorithm was an attempt to produce dense motion estimation. Yet because the method is easily applied to a subset of the points in the input image, it has become an important sparse technique. The LK algorithm can be applied in a sparse context because it relies only on local information that is derived from some small window surrounding each of the points of interest. This is in contrast to the intrinsically global nature of the 3DRS. The disadvantage of using small local windows in Lucas-Kanade is that large motions can move points outside of the local window and thus become impossible for the algorithm to find. This problem led to development of the "pyramidal" LK algorithm [2], which tracks starting from highest level of an image pyramid (lowest detail) and working
Figure 3.2: Pyramid Lucas-Kanade optical flow: running optical flow at the top of the pyramid first mitigates the problems caused by large motion vector. The motion estimate from the preceding level is taken as the starting point for estimating motion at the next layer down.

down to lower levels (finer detail). Tracking over image pyramids allows large motions to be caught by local windows.

Let \( I \) and \( J \) be two 2D grayscaled images. The two quantities \( I(X) = I(x, y) \) and \( J(X) = J(x, y) \) are then the grayscale value of the two images at the location \( X = [x \ y]^T \), where \( x \) and \( y \) are the two pixel coordinates of a generic image point \( X \). Consider a point \( u = [u_x \ u_y]^T \) on the first image \( I \). The goal of feature tracking is to find the location \( v = u + d = [u_x + d_x \ u_y + d_y]^T \) on the second image \( J \) such as \( I(u) \) and \( J(v) \) are "similar". The vector \( d = [d_x \ d_y]^T \) is the image velocity at \( X \), also known as optical flow at \( X \). Let \( w_x \) and \( w_y \) be two integers. Then the image velocity \( d \) is defined as the vector that minimises the residual function \( \varepsilon \) defined as follows:

\[
\varepsilon(d) = \varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x+d_x, y+d_y))^2
\]

The above definition similarity function is measured on an image neighbourhood of size \((2w_x + 1) \times (2w_y + 1)\) which is also called as integration window. Typically \( w_x \) and \( w_y \) ranges in 2 to 7.

The image pyramid is constructed for image \( I \) and \( J \). Let \( I^0 = I \) be the zeroth level image. The pyramid representation for pyramidal height \( L_m + 1 \) is then build in a recursive fashion: compute \( I^1 \) from \( I^0 \), then \( I^2 \) from \( I^1 \) and so on till calculating \( I^m \) from \( I^{m-1} \) as shown in Figure 3.2. With increase in image level, size is reduced by half. Anti-aliasing filter is used before sub-sampling the image to avoid artefacts. The central motivation behind pyramidal representation is to be able to handle large pixel motions (larger than the integration window sizes \( w_x \) and \( w_y \)). Therefore the pyramid height (\( L_m \)) should also be picked appropriately according to the maximum expected optical flow in the image.

For \( L = 0 \ldots L_m \), define \( u^L = \frac{u}{2^L} = [u_x^L \ u_y^L] \), the corresponding coordinate of the point \( u \) on the pyramidal images \( I^L \).

The overall pyramidal tracking algorithm proceeds as follows:

- First, the optical flow is computed at the deepest pyramid level \( L_m \).
- Then, the result of that computation is propagated to the upper level \( L_m - 1 \) in a form of an initial guess for the pixel displacement (at level \( L_m - 1 \)).
- Given that initial guess, the refined optical flow is computed at level \( L_m - 1 \), and the result is propagated to level \( L_m - 2 \) and so on up to the level 0 (the original image).
Let us consider the recursive operation between two generic levels $L + 1$ and $L$ more mathematically. Assume that an initial guess for optical flow at level $L$, $g^L = [g^L_x g^L_y]^T$ is available from the computations done from level $L_m$ to level $L + 1$. Then, in order to compute the optical flow at level $L$, it is necessary to find the residual pixel displacement vector $d^L = [d^L_x d^L_y]^T$ that minimizes the new image matching error function $\varepsilon_L$:

$$\varepsilon_L = \varepsilon_L(d^L_x, d^L_y) = \sum_{x = u^L_x - w_x}^{u^L_x + w_x} \sum_{y = u^L_y - w_y}^{u^L_y + w_y} (I^L(x, y) - J^L(x + g^L_x + d^L_x, y + g^L_y + d^L_y))^2 \quad (3.2)$$

The window size of the integration is of constant size $(2w_x + 1) \times (2w_y + 1)$ for all values of $L$. The initial guess flow vector $g^L$ is used to pre-translate the image patch in the second image $J$. That way, the residual flow vector $d^L = [d^L_x d^L_y]^T$ is small and therefore easy to compute through a standard Lucas Kanade step [2]. The result of this computation is propagated to the next level $L - 1$ by passing the new initial guess $g^{L-1}$ of expression:

$$g^{L-1} = 2(g^L + d^L) \quad (3.3)$$

The next level optical flow residual vector $d^{L-1}$ is then computed through the same procedure. This vector, computed by optical flow computation (described in [2]), minimizes the functional $\varepsilon^{L-1}(d^{L-1})$ (equation 3.2). This procedure goes on until the finest image resolution is reached ($L = 0$). The algorithm is initialized by setting the initial guess for level $L_m$ to zero (no initial guess is available at the deepest level of the pyramid).

$$g^{L_m} = [0 0]^T \quad (3.4)$$

The final optical flow solution $d$ (refer to equation 3.1) is then available after the finest optical flow computation as $d = g^0 + d^0$.

The clear advantage of a pyramidal implementation is that each residual optical flow vector $d^L$ can be kept very small while computing a large overall pixel displacement vector $d$. Assuming that each elementary optical flow computation step can handle pixel motions up to $d_{max}$, then the overall pixel motion that the pyramidal implementation can handle becomes $d_{max/ final} = (2^{L_m + 1} - 1)d_{max}$. For example, for a pyramid depth of $L_m = 3$, this means a maximum pixel displacement gain of 15! This enables large pixel motions, while keeping the size of the integration window relatively small.

### 3.5 Selecting the features to track

Typically in OpenCV, cvGoodFeaturesToTrack() and cvCalcOpticalFlowPyrLK() are used in combination for motion tracking. cvGoodFeaturesToTrack() function gives the feature points which are "good to track" by cvCalcOpticalFlowPyrLK(). In Pyramidal LK algorithm, the central step for motion tracking is the computation of an optical flow vector $\mathbf{\eta}^k = G^{-1}\mathbf{b}_k$ [2], where $G$ and $\mathbf{b}_k$ are Spatial gradient matrix and Image mismatch vector respectively. For calculating a value of $\mathbf{\eta}^k$, the value of $G$ matrix is required to be invertible, or in other words, the minimum eigenvalue of $G$ must be large enough (larger than a threshold). Thus considering this as a characterising feature to distinguish the pixels that are "easy to track" are detected by the function cvGoodFeaturesToTrack() [25]. These pixels or features according to LK algorithm are then provided to cvCalcOpticalFlowPyrLK() function for tracking in the next frame.
3.6 Patch tracking

The motion tracking system implemented using LK method is called patch tracking. The patch tracking system is implemented in C++ using OpenCV 2.0 library. In this system, various feature points for tracking are obtained by cvGoodFeaturesToTrack() function in OpenCV. The feature points lying out of the skinmap are discarded keeping the features which are in the skin area and which are to be tracked in successive frames. The cvGoodFeaturesToTrack() function is executed on each frame to capture appearance of any new trackable feature. Given the trackable features in a frame of a video, the cvCalcOpticalFlowPyrLK() function can find the location of those features in the successive frame. The function also notifies if the feature is no longer trackable.

We consider a square, called patch, of size patchSize with trackable feature point at its centre. When such a square patch is moving through spatial-temporal domain, it marks a space called trace. Each trace has a starting point, traceStart, and the ending point, traceEnd, along the time. The start point traceStart is the point when cvGoodFeaturesToTrack() function declares a feature of a patch corresponding to the given trace as trackable. The end point, patchEnd, is the point when cvCalcOpticalFlowPyrLK() declares the same feature as non-trackable. The length of a trace from traceStart to traceEnd along the time axis is called traceLength. The above mentioned terminologies can be visualised in Figure 3.3. Each trace is further divided into three traces viz. R,G and B trace. These traces are obtained by averaging over the patch area ( at locations defined by the trace ) in RGB subimages. The R,G,B traces obtained have same traceStart, traceEnd and so traceLength. In short dynamic variation in average R,G,B values over the trace is given by R,G,B traces. The data for each trace is dumped into a file on its termination. These files are then read by a Matlab program for further processing.

To understand the dynamic nature of traces we construct a diagram called trace visualisation (traceVis). In traceVis traces are visualised along the time axis. The traces are displayed from top to bottom in the sequence they are declared non-trackable by cvCalcOpticalFlowPyrLK() function. The information such as traceStart, traceEnd, traceLength of each trace and dynamic appearance-disappearance of various traces can be obtained from traceVis. The traceVis diagrams constructed
3.7 BSS with patch tracking

In FD-BSS and FD-BSS-IPPP, average over a big ROI interest is taken. An average taken over successive frames gives RGB trace. In case of patch tracking, however, we have many small ROIs giving us a choice of either combining all small ROIs together to get a final RGB trace or analyse each ROI independently. We call first approach as Patch Tracking-Blind Source Separation with Improved Pre and Post Processing (PT-BSS-IPPP) while the later is called Patch Tracking-Blind Source Separation with Improved Pre and Post Processing for Independent Traces (PT-BSS-IPPP-IT). In both cases, PT-BSS-IPPP and PT-BSS-IPPP-IT, patchSize is initialised to 8.

In PT-BSS-IPPP, average over all the pixels in all the patches over R,G and B sub-images of a given frame is calculated to give value in RGB trace (Similar to Figure 2.2 except that big ROI is replaced by many small patches on face). Applying same process on successive frames gives RGB trace. The pre and post processing applied on the RGB trace is same as FD-BSS-IPPP. Thus, we can say PT-BSS-IPPP is motion tracking improved (block 2 in Figure 2.1 improved) version of FD-BSS-IPPP.

In PT-BSS-IPPP-IT, RGB traces for each and every patch is managed independently. Instead of creating complete RGB trace for a complete video sequence for further processing, small RGB trace over a Window are constructed. Individual traces are constructed by taking average over R,G and B sub-images for the patch area in successive frames. The traces can be of any length depending on the motion of a subject, as we saw in Figure 3.4. To calculate the HR signal with each WindowOverlap step, we take all the traces which survive for traceLength = Window. Let m be the number of traces survived. The ICA is applied to each of these m traces and out of the separated components, the component with the maximum peak is selected for further processing. Let these selected components are available in $X$. The signals $X(0)$ to $X(m)$ are combined to form a trace to give a final signal using

![Figure 3.4](image1.png)  ![Figure 3.4](image2.png)

(a) traceVis for translation02  (b) traceVis for rotation02

Figure 3.4: The Figure 3.4a illustrates the traceVis for translation02 over 1800 frames. Due to the translational motion traces survive for longer duration. Many traces at the bottom of the figure can be seen which survives over complete length of a video. The Figure 3.4b illustrates the traceVis for rotation02 over 1200 frames. Due to rotational motion traces survive for short duration (depends on the rate of rotation).
energy weighting as follows:

\[ X' = \frac{\sum_{n=0}^{m} \frac{X(n)}{\text{Energy}(X(n))}}{\sum_{n=0}^{m} \frac{1}{\text{Energy}(X(n))}} \]  

(3.5)

The Energy of a signal is calculated as:

\[ \text{Energy}(x) = \sum_{n=0}^{N} x(n)^2 \]  

(3.6)

where \( N \) is the length of a signal, which in our case, is \( \text{Window} \). The HR signal is then extracted from \( X' \) by taking FFT and finding the highest peak. The logic behind the energy weighting is that the signal with less variation over the time has less energy. The pulse variation on a trace by the BVP signal is of low amplitude while the variation caused by a motion of a subject is of high amplitude. Thus, the signal with less energy, which is likely to have more accurate BVP information, is given higher weight as compared to the others.

### 3.8 Results

In this section, first we compare the performance of FD-BSS-IPPP, SM-BSS-IPPP and PT-BSS-IPPP to verify the improvement in performance with the change in motion tracking. In Figure 3.5, a bar graph for SNR and RMSE values for various test videos is plotted. In Figure 3.5a, it can be noticed that SM-BSS-IPPP and PT-BSS-IPPP has improved the SNR for many video sequences. In some cases improvement is by large amount while in some others it is insignificant. In Figure 3.5b, unless we count improvement in rotation00 there is no other significant improvement in RMSE value can be observed. By the comparison of FD-BSS-IPPP and SM-BSS-IPPP in terms of SNR and RMSE, we can conclude that SM-BSS-IPPP performs better. The improvement from FD-BSS-IPPP to PT-BSS-IPPP was expected to be by large margin considering the elimination of non-skin pixels using a skinmap and improvement in the motion tracking. A reason for the poor performance is the ICA not being able to separate a BVP signal satisfactorily under large motion.

Next, we compare the performance of PT-BSS-IPPP and PT-BSS-IPPP-IT. In section 3.6, we have seen that the length of the traces depends on the type of motion. From Figure 3.4, we know that depending on the type of motion in a video maximum length over which the patches survive varies. Thus, PT-BSS-IPPP-IT can not be used for processing the video sequences in which patches can not survive over \( \text{Window} = 30\text{Sec} \). Thus we compare performance of PT-BSS-IPPP and PT-BSS-IPPP-IT by comparing their performance on color01,02 and translation00,01 and 02. In Figure 3.6, a bar graph for SNR and RMSE values is illustrated for these videos. In Figure 3.6a, we see that PT-BSS-IPPP-IT has improved the SNR except for translation00 sequence. In case of translation00 sequence an independent component with the peak for motion got selected as it had higher peak value than the peak due to the HR. This also results in poor RMSE value for translation00 sequence in Figure 3.5a. For other videos PT-BSS-IPPP-IT performs better than PT-BSS-IPPP.

If we could use PT-BSS-IPPP-IT with reduced \( \text{Window size} \) (according to FD-BSS it is 30sec i.e. 600 frames) we could use it for videos which do not have long surviving traces (e.g. rotational videos). Thus, the effect of variation of a \( \text{Window size} \) on the performance of PT-BSS-IPPP and PT-BSS-IPPP-IT is investigated. As the performance of PT-BSS-IPPP and PT-BSS-IPPP-IT depends on how well ICA separates the independent components, we are mainly concerned with the performance of ICA with variation in \( \text{Window size} \). The ICA being a computational method which increase the mutual statistical independence of the non-Gaussian source signals, its performance degrades when it has less
3.8 Results

(a) SNR comparison of FD-BSS-IPPP, SM-BSS-IPPP and PT-BSS-IPPP

(b) RMSE comparison of FD-BSS-IPPP, SM-BSS-IPPP and PT-BSS-IPPP

Figure 3.5: Performance comparison of FD-BSS-IPPP, SM-BSS-IPPP and PT-BSS-IPPP.

(a) SNR comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT

(b) RMSE comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT

Figure 3.6: Performance comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT.
3.9 Conclusion

In this chapter we have introduced new method for selecting and tracking the ROI (improvement in block 2 of Figure 2.1). We introduced PT-BSS-IPPP and PT-BSS-IPPP-IT which are the motion improved version of FD-BSS-IPPP, introduced in the previous chapter. In the last section, we saw the results of PT-BSS-IPPP and PT-BSS-IPPP-IT. We saw that PT-BSS-IPPP-IT performs better than PT-BSS-IPPP. However, PT-BSS-IPPP-IT can not be used practically unless the motion of a subject is restricted. Also, a large Window size is necessary to get the best results from ICA. On the other hand, increase in Window size gives larger initial delay, slow responsiveness to changes in HR variations and requires limited motion of a subject. PT-BSS-IPPP does not have any limitation on the type of motion of a subject. However, with large motion ICA fails to give better separation of independent components. ICA can separate $n$ independent signal from a mixture of $n$ signals. It can not separate $n$ signal from $m$ ($<n$) mixture signals. The RGB traces obtained from a video are affected by many conditions viz. BVP signal, intensity variations, colour variations, motion of a subject etc. For further developments in the motion robust algorithm, stronger signal processing is needed. We consider PT-BSS-IPPP as a benchmark for comparing performance improvement in development of the algorithm.

Figure 3.7: Performance comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT.

(a) SNR comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT

(b) RMSE comparison of PT-BSS-IPPP and PT-BSS-IPPP-IT

The test on color01 (for which PT-BSS-IPPP-IT has highest SNR (Figure 3.6a)) and translation02 (for which PT-BSS-IPPP has highest SNR (Figure 3.6a)) is performed. In Figure 3.7, four curves in each graph corresponds to the SNR and the RMSE for color01 and translation02 are illustrated. Along the x-axis, the Window size is varied from 100 to 600 frames in a step of 100 frames. In general, with increase in Window size available data for ICA analysis increase. It is evident from increase in SNR and decrease in RMSE curves that ICA performs better with larger Window size. It is also visible from Figure 3.7b that maximum performance of an algorithm is reached early in case of color01 than translation02. As there is no motion in color01, window length required for ICA to give good results is less as compared to translation02 in which translational motion is present.
Improving the signal extraction from Region of Interest

In previous chapters, we introduced FD-BSS and improved it by making suitable changes in pre and post-processing and motion tracking (block 2, 5 and 7 of flowchart in Figure 2.1). In the last chapter, we saw that with high degree of motion, ICA can not help to separate the independent components. A new technique is needed to extract the BVP signal. The improvements in ROI selection and tracking (block 2) are important and can be used in further development. The next step in flowchart after ROI selection and tracking (block 2) is the extract the signal from ROI (block 3 in Figure 2.1) which defines the way raw traces are extracted from a video. Improvements in this step are valuable for performance of the algorithm as it is the raw traces which go under further processing to yield the HR signal. FD-BSS extracts raw signal by dividing a colour image in RGB sub-images. In RGB colour space, colour and intensity information is intermixed in RGB channels. Apart from RGB colour space, there are other interesting colour models which separates the colour information from the intensity information in an image. A use of colour model which can separate intensity variations is theoretically attractive for extracting better raw traces. Next, we investigate two important colour models viz. HSI and YCbCr used for the HR extraction.

4.1 HSI : Hue-saturation-intensity

The transformation from RGB to HSI colour space is illustrated in Figure 4.1a. The projection of RGB cube along the main diagonal (line connecting black \(\{0,0,0\}\) and white \(\{1,1,1\}\)) is illustrated. In the projection, applying radial coordinate system gives the HSI colour model. Such a use of radial coordinate system is illustrated in Figure 4.1b. In the figure, red is considered as a reference to measure an anti-clockwise angle to represent a Hue i.e. a colour. The radial distance of a point from the centre Intensity axis represents Saturation. The HSI colour model is very important and attractive for image processing applications because it represents colours similar to how the human eye senses the colours. When humans see a colour object, we describe it by its hue, saturation and intensity.

The Hue component describes the colour in the form of an angle within \([0,360]\) range. For example, 0 degree means red, 120 means green 240 means blue, 60 degrees is yellow and 300 degrees is magenta. The saturation component represents how much the colour is polluted with white colour. The range of the S component is \([0,1]\). The saturation value equal to 1 represents a pure colour. The Intensity range is between \([0,1]\), where 0 means black and 1 means white. As the Figure 4.1b illustrates, hue is more meaningful when saturation approaches 1 and less meaningful when saturation approaches 0 or
4.1 HSI: Hue-saturation-intensity

(a) Above figure illustrated the transformation from RGB to HSI colour space. R,G,B are red, green and blue colours while Y,C,M are yellow, cyan and magenta colours, respectively.

(b) HSI colour space.

Figure 4.1: HSI colour model.

(a) AC amplitude of Hue(red) and Saturation(blue) traces extracted from video color01 are illustrated in above figure. Hue trace shows strong BVP signal. Saturation trace carries BVP information too however, it is weak and noisy compared to Hue.

(b) FFT of the Hue and Saturation traces in illustrated in Figure 4.2a

Figure 4.2: Analysis of AC Hue and Saturation traces extracted from color01.rgb video sequence.
4.1 HSI : Hue-saturation-intensity

(a) Above figure illustrates the HS plane. R is considered as a reference for measuring an angle in anti-clockwise direction. In the figure, a small black line in the skin pixels range, along which hue and saturation changes with BVP is illustrated.

(b) Closer look at hue and saturation trace shows that, peak-to-peak AC amplitude of hue and saturation for colour01 video is 0.0003 and 0.0006. If amplitude of hue is converted to an angle, it is close 0.12 degree. Saturation value can be affected by the change in illumination. Hue is the only component which does not have influence of illumination changes and it is very small. This shows the magnitude of the problem in signal extraction with motion.

Figure 4.3: HS trace analysis.

The HSI colour model is interesting for the HR analysis because it decouples the intensity component from the colour-carrying information (hue and saturation) in a colour image. The pixels in the range \( H = [0 \, 50] \) and \( S = [0.23 \, 0.68] \) covers all types of skin colours [23]. This range is mapped on HS plane in Figure 4.3a. As the effect of intensity is minimal in HS plane, better results can be expected with varying illumination of a skin patch because of a motion. In Section 1.3 we saw, the light received by the receiver has a combination of specular reflection and diffused light (Figure 1.6). The diffused light components intensity depends on the BVP in the skin vasculature. Assume for some condition of BVP, diffused light is more intense. In such a case, we can imagine the diffused light adding to the specular reflection causes small changes in the hue and saturation of a skin colour. In short, a small change in hue component is expected with change in blood volume in vasculature.

To verify this hypothesis we implemented a program with the patch tracking system as mentioned in the previous chapter and some changes in signal extraction (block 3). The extract the signal from ROI step in the flowchart (Figure 2.1) is replaced by extract average HSI from ROI i.e. to take the average over pixels in ROI, frame is divided into HSI sub-images instead of RGB sub-images. The AC amplitude of the hue and saturation trace for color01 is illustrated in Figure 4.2a. In the trace hue is scaled from 0 to 1. With each heartbeat, when heart pumps blood to the periphery, skin hue becomes more red causing hue to drop. A closer look at hue and saturation trace, in Figure 4.3b, shows that the change in amplitude of hue with BVP is roughly of magnitude 0.0003 i.e. 0.12 degrees. This gives us an idea about how small the hue changes are with the changes in BVP. Comparatively saturation magnitude is bigger, viz. 0.0006. The saturation waveform carries BVP signal too (Figure 4.2b). However it should be noticed that the plot is for video color01, in which there is no motion. Saturation component is affected by the illumination changes. From above observation, for some hue of a subject’s skin, with BVP, considerably large changes in saturation are accompanied by the small angular variations in hue. Though in practice such a variation curve would be very small, for an
4.2 Results with HSI

In this section, the performance of PT-H with PT-BSS-IPPP is compared to evaluate the effect of HSI colour model on the performance improvement. In Figure 4.4, a bar graph for SNR and RMSE values for various test videos is plotted. As can be seen from the figure, HSI performs as good as PT-BSS-IPPP when motion is limited, e.g. color01, color02 and translation02. With large motion, large variations in hue trace is observed which corrupts the BVP signal. The pre-processing in PT-H includes only band pass filtering in [40, 220] bpm which is unable to eliminate the noise due to motion as it falls in the same range.

4.3 YCbCr

YCbCr is the next important colour model which separates the intensity information from the colour information. YCbCr is a family of colour spaces used as a part of the colour image pipeline in video and digital photography systems. Y is the luminance component and Cb and Cr are the blue-difference and red-difference chroma components. YCbCr is not an absolute colour space, it is a way of encoding
RGB information. In general, YCbCr is used to separate out a luminance signal (Y) and two chroma components (Cb and Cr).

YCbCr values can be obtained by a linear combination of RGB values. In our case, OpenCV library is used to convert RGB image into YCbCr image. The linear conversion from RGB to YCbCr as defined in OpenCV is as follows:

\[
Y = 0.299 \times R + 0.587 \times G + 0.114 \times B
\]
\[
Cb = (B - Y) \times 0.564 + 128
\]
\[
Cr = (R - Y) \times 0.713 + 128
\]

(4.1)

The constant value, 128, in CbCr is to scale the values in the range [0 255]. Neglecting this scaling, the YCbCr equation is as follows:

\[
Y = 0.299 \times R + 0.587 \times G + 0.114 \times B
\]
\[
Cb = -0.168 \times R - 0.331 \times G + 0.499 \times B
\]
\[
Cr = 0.499 \times R - 0.418 \times G - 0.081 \times B
\]

(4.2)

As calculation of YCbCr has more involved mathematics it was difficult to predict the nature of waveforms before hand and the signal which would carry the BVP signal. We implemented a program similar to the one for PT-H in C++ replacing extract signal from ROI in Figure 2.1 by extract average YCbCr from ROI i.e to take the average over pixels in ROI, frame is divided into YCbCr sub-images instead of RGB sub-images. The extracted traces are read in Matlab for further processing from the file storing these traces. Considering the knowledge that Cb and Cr carry colour information we examine the AC amplitudes of Cb and Cr traces for color01 video sequence in Figure 4.5a. At first it appears that both signals carry a weak BVP signal but a closer look and FFT analysis reveals that Cb carries stronger BVP signal than Cr. We select Cb component for the HR analysis. This algorithms is called Patch Tracking - Cb (PT-Cb). Also, considering the anti-phase nature of Cb and Cr traces, the addition signal was investigated which revealed to be carrying the stronger BVP signal than Cb and Cr.

An explanation for the BVP signal in Cb and Cr can be given from Equation 4.2. The YCbCr vectors in Equation 4.2 are visualised in Figure 4.6a. The CbCr plane is parallel to the HS plane. The mapping of the skin pixels range in HS plane by CbCr, as illustrated in Figure 4.3a, is illustrated in Figure 4.6b along with the hue and saturation change curve caused by BVP changes. The mapping of this curve on Cb and Cr is illustrated by dotted lines in the figure. The anti-phase nature of Cb and Cr curve can be explained from the Figure 4.6b.

The CbCr signals, depending on the location of the trace (shown in Figure 4.6b), carry the BVP information of different strength. This fact can be exploited to get a better BVP signal. For extracting a signal from Cb and Cr traces, a pre-processing algorithms is developed. Using this algorithm the BVP signal can be constructed. The Algorithm 1 takes two in-phase signal (Cb and -Cr). It shapes and stretches one signal to match the major variation in amplitude (caused by saturation change) of the other. Hence, when we take a difference between these two signals, the BVP signal which is of different strength on these signals, gets separated. Let us call the final signal obtained from the algorithm as HB. The HB signal is then analysed to extract the HR. We call this algorithm Patch Tracking - Cb Vs Cr (PT-Cb-Cr). In both, PT-Cb and PT-Cb-Cr, Window and WindowOverlap are of size 256 and 8 respectively same as in PT-H. The band pass filter in Algorithm 1 is filters the frequencies in the HR range of 40-220 bpm. No post-processing was applied.
4.3 YCbCr

(a) AC amplitude of Cb (blue) and Cr (black) traces extracted from video color01 are illustrated in above figure. Addition signal of Cb and Cr shows the stronger BVP signal.

(b) FFT of the Cb, Cr and Cb+Cr AC traces illustrated in Figure 4.5a

Figure 4.5: Analysis of AC Cb and Cr traces extracted from color01.rgb video sequence. The opposite nature of Cb and Cr can be noticed in Figure 4.5a. To see the effect of addition of Cb and Cr, Cb+Cr traces is also analysed in above figure.

(a) The YCbCr vectors are illustrated in 3D. Cb vector extended in negative direction is also illustrated.

(b) Mapping of the sample hue and saturation change curve (from Figure 4.3a) caused by BVP changes is shown in above figure. Dotted lines shows the mapping of the curve onto the Cb and Cr axes.

Figure 4.6: Analysis of YCbCr model and skin pixels mapping.
4.4 Results with YCbCr

Algorithm 1 The function for constructing a trace by combining $X = Cb$ and $Y = -Cr$ trace. The Window and WindowOverlap used in this algorithm are different than the one used in pre-processing. The Window and WindowOverlap used in this algorithm are 32 and 4 frames, respectively.

getHB function:
begin
    $X f = \text{bpf}(X)$;
    $Y f = \text{bpf}(Y)$;
    for $i := 1$ to Video_Length step WindowOverlap do
        $X t = X f(\text{Window})$;
        $Y t = Y f(\text{Window})$;
        $X t = X t - \text{mean}(X t)$;
        $Y t = Y t - \text{mean}(Y t)$;
        $Y t = \frac{\text{std}(X t)}{\text{std}(Y t)} \times Y t$;
        $H B t = X t - Y t$;
        $H B(\text{Window}) = H B(\text{Window}) + H B t$;
        $H B t = \frac{H B t}{\text{std}(H B t)}$;
        $H B(\text{Window}) = H B t$;
    od
    $H B = \text{bpf}(H B)$
end

4.4 Results with YCbCr

In this section we compare the performance of PT-Cb and PT-Cb-Cr with PT-BSS-IPPP. In Figure 4.7, bar graphs for SNR and RMSE values for various test videos is illustrated. In Figure 4.7a, it can be observed that in most of the cases PT-Cb-Cr gives better SNR than PT-Cb. This was expected considering the strong BVP component in HB signal than in Cb. The most important thing to note here is that PT-Cb-Cr performs as good as PT-BSS-IPPP or better in some cases viz. rotation03 and translation00.

Considering Figure 4.7b, it can be observed that in terms of RMSE, PT-Cb-Cr performs better than PT-Cb in all cases except for translation02. In case of translation02, PT-Cb-Cr would have performed better, however, at few time instant peak corresponding to the motion dominated the fundamental peak causing large deviation from RHRT. As we have not used any post-processing yet, this problem persists. Except for translation02, significant improvement is noticed in rotation02, rotation01 and translation00 in terms of RMSE for PT-Cb-Cr over PT-BSS-IPPP.

4.5 RGB

In the last section we saw that YCbCr gave better results than HSI. We also investigated the reason for better performance of YCbCr. In this section we dive into details of absorption of light by Oxyhemoglobin and De-Oxyhemoglobin to achieve further improvement. Hemoglobin (Hb) is the iron-containing oxygen-transport metalloprotein in the red blood cells (RBCs). Hb in the blood is what transports oxygen from the lungs to the rest of the body (i.e. the tissues) where it releases the oxygen for cell use, and collects carbon dioxide to bring it back to the lungs. The name heamoglobin is derived from the words heme and globin, reflecting the fact that each subunit of Hb is a globular
protein with an embedded heme (or haem) group. Each heme group contains one iron atom, that can bind one oxygen molecule. Oxyhemoglobin (OHb) is formed during physiological respiration when oxygen binds to the heme component of the protein Hb in RBCs. The oxygen then travels through the blood stream to be dropped off at cells where it is utilized. De-Oxyhemoglobin (DHb) is the form of Hb without the bound oxygen. The absorption spectrum of light for OHb and DHb differ and it is as shown in Figure 4.8a [31]. In section 1.1 we saw that skin has the congregation of blood vessels in Dermis and Hypodermis layer. These blood vessels can be arteries which carry oxygenated blood from heart to other body parts or veins which carry de-oxygenated blood from body parts to heart. Thus arteries have more amount of OHb while veins have DHb. When light incidents on a patch of skin it interacts with the mesh of vasulature which has arteries and veins. Thus light is absorbed by OHb as well as DHb. A good approximation, considering the equal distribution of arteries and veins in skin layers, could be that equal amount of light is incident on OHb as well as DHb. In such a case the absorptivity seen by a light falling on the skin can be approximated as the average absorptivity by OHb and DHb. Such an absorptivity curve is illustrated in Figure 4.8b by a dotted line. As the light absorbed by other tissue (muscles and bones etc.) remain constant we can safely eliminate it from consideration. The graph in Figure 4.9 illustrates the variation of AC amplitude of a PPG signal against the wavelength of a light [15] confirms that this approximation is valid.

R, G and B are the principle colour components, combination of which can give any other colour. The sensitivity of the red, green and blue sensors of the uEye camera used for capturing videos is illustrated in Figure 4.8. In the same figure, the dashed curve represents the absorptivity of the skin for different wavelength. Higher the absorptivity of a light for a certain wavelength, larger is the amplitude of AC variations for the same wavelength. This can be verified by comparing absorptivity curve in Figure 4.8 and the AC amplitude curve in Figure 4.9.

The major area under absorptivity curve of the green component overlaps the sensitivity curve of the camera for the green light. This results in strong BVP signal in the green trace (Figure 4.10b). The blue and the red components carry BVP signal which is comparatively weak. It is clear from the Figure 4.10b that for blue and the red components, if compared the HR peak level with the rest of the spectrum level, red component has higher SNR. However, the noise floor of the blue component closely matches with the noise floor of the green component. Overall variation in the energy of RGB components can also be noticed from Figure 4.10b. The descending order of energy in spectrum can
4.5 RGB

(a) Above figure illustrates the absorptivity of light by (1) Oxyhemoglobin and (2) De-Oxyhemoglobin for adult hemoglobin (Hb A) and fetal hemoglobin (Hb F) as given in [31]. Absorptivity is expressed in \( L \cdot \text{mmol}^{-1} \cdot \text{cm}^{-1} \) [31]. An approximation which can be used for light absorption by the skin is illustrated by the dotted line in above figure.

(b) The mapping of the RGB sensors sensitivity of the uEye camera on the light absorptivity curve for skin is illustrated in above figure. It is clear from this figure that a green light has the highest absorptivity and sensitivity by the camera sensor.

Figure 4.8: Absorptivity curves

Figure 4.9: Above figure illustrates the variation in the AC amplitude of a PPG signal along the wavelength of a source. [15]
4.5 RGB

(a) AC RGB traces for colour01 video sequence is illustrated in above figure. It should be noticed that all traces have same major variations which are caused by intensity changes. The green component shows strong BVP signal as compared to R and B.

(b) FFT spectrum for RGB traces illustrated in Figure 4.10a

Figure 4.10: Analysis of normalised RGB traces extracted from color01.rgb video sequence.

be given as: green, blue, red. These two observation can be explained by the IR-cut-filter at 650nm. The red component does not have as much energy as in the others because the higher frequencies in its main lobe of sensitivity are filtered out. Considering above observations, if we take subtraction of two spectra, say green and blue, we can expect a cleaner spectrogram as the matched noise in rest of the part of spectrum gets nullified. Moreover, as we discussed before, RGB traces carry same information corresponding to the specular reflection. With this idea, we plot three traces corresponding to the signal R-G, R-B and G-B in Figure 4.11a. These particular combinations makes each trace pair with the other two and gives orthogonal vectors to the intensity axis. A frequency analysis of these traces reveals significant information. In Figures 4.11b frequency spectrum for AC waveforms (Equation 2.1) G-B, R-G and R-B are shown.

From the visual analysis of these spectrum’s it is clear that R-G and G-B carry stronger BVP signal while R-B’s spectrum do not show any peak corresponding to the BVP. Comparing the spectrum of the AC waveforms RGB traces illustrated in Figure 4.10b with the spectrum of new traces in Figure 4.11b shows that new traces have better SNR. If we compare the spectrum of R-G and G-B in Figure 4.11b, it can be noticed that G-B has the stronger BVP signal than R-G. Third harmonic corresponding to the HR is stronger in G-B trace than in R-G trace. All these results have their explanation in Figure 4.8b. The light in green and blue components range have absorptivity very close to each other. Whatever changes green channel goes through, blue channel also goes through them but with little less impact as it has lower absorptivity. Due to this, blue trace follows the same changes as in green trace except that the green trace has the stronger AC variations of the BVP. Thus, when blue trace is subtracted from green trace, we are subtracting all the common variations and only the distinct variation on green trace, which corresponds to the BVP, are separated. Similar explanation can be given for trace R-G. Less energy in the red component and the separation of noise floor of red and green visible in Figure 4.10b results in comparatively less SNR in R-G trace. If noticed closely R-G is close to the inverted G-B trace. To conclude, the R-G trace is not performing as good as G-B because the variation in the green channel are reflected more faithfully by the blue channel than the red channel.

Thinking on the similar lines of YCbCr, if we construct an orthogonal trace to R-G and G-B we can
Traces obtained by taking substration of RGB traces are plotted. In above figure blue, green and red traces respectively corresponds to the traces obtained by normalising G-B, R-G and R-B traces.

Figure 4.11: Analysis of AC waveforms obtained by taking the difference within sub-channels. The pairs are selected such that each component is paired with the other two.

get results as good as YCbCr or better. A complete vector R-G and G-B can be given as R-G+0×B and 0×R+G-B. In vector form they can be written as \([1 \ -1 \ 0]^T\) and \([0 \ 1 \ -1]^T\) respectively. We need to find orthogonal vectors to these vectors such that the addition of coefficients of those vectors is zero. First, we consider vector \([1 \ -1 \ 0]^T\). Let \([a \ b \ c]^T\) be the orthogonal vector such that \(a+b+c=0\). We know that product of the orthogonal vectors is zero. Using this property we get \(a - b = 0\). Solving above two equations we get the orthogonal vector as \([1 \ 1 \ -2]^T\). Similarly, we find the orthogonal vector for \([0 \ 1 \ -1]^T\) as \([-2 \ 1 \ 1]^T\). In terms of RGB equation, these vectors can be written as R+G-2B and -2R+G+B respectively. The vectors we discussed till now i.e. R-G, G-B, R-B, R+G-2B and -2R+G+B are illustrated in Figure 4.13a. The traces corresponding to R+G-2B and -2R+G+B are as shown in Figure 4.12 and their respective power spectrums are in Figure 4.12b. From Figures 4.12b it is clear that R+G-2B has the strong BVP signal as compared to -2R+G+B as the first shows the second and third harmonic of the HR while the later does not. We use the getHB (Algorithm 1) function to get the HB signal from these two sets of orthogonal signals. The algorithm which uses first pair, \(\left\{[1 \ -1 \ 0]^T, [1 \ 1 \ -2]^T\right\}\), is called Patch Tracking - RG Vs RG2B (PT-RG-RG2B) while the algorithm which uses second pair, \(\left\{[0 \ 1 \ -1]^T, [-2 \ 1 \ 1]^T\right\}\) is called Patch Tracking - GB Vs 2RGB (PT-GB-2RGB). The getHB function simply stretches one signal to match the variation in amplitude of the other and finally combine the two signals together to give stronger resultant signal.

The vectors, R-G, G-B, R-G, R+G-2B and -2R+G+B are visualized in Figure 4.13a. R-B vector falls in the skin pixels region and it is parallel to the saturation variations. Thus, signal on this vector is effected by the illumination changes. The final two vector orthogonal pairs are illustrated in Figure 4.13b. The projection of the trace of HS variation with the BVP changes is illustrated on these pairs. The G-B pair, as can be seen from Figure 4.11b, has the highest SNR as it is least effected by the variations in the saturation value.

Similar to PT-Cb-Cr a C++ code is implemented for PT-RG-RG2B and PT-GB-2RGB using patch tracking system to extract RGB traces from various video sequences. The pre-processing code is implemented in Matlab. The Algorithm 1, same as used in PT-Cb-Cr, is implemented to extract
4.5 RGB

(a) The AC traces for R+G-2B and -2R+G+B are illustrated.

(b) FFT spectrum for the traces R+G-2B and -2R+G+B shown in Figure 4.12a

Figure 4.12: Analysis of the AC R+G-2B and -2R+G+B traces.

(a) R-G, G-B, R-B, R+G-2B and -2R+G+B vectors are visualized.

(b) Mapping of the hue and saturation change trace due to the change in BVP on to the few vectors is illustrated. This shows that the large the inclination of the of a vector, larger is the footprint on the vector.

Figure 4.13: Vectors analysis.
4.6 Results

In this section we compare the results of PT-RG-RG2B and PT-GB-2RGB with PT-Cb-Cr and PT-BSS-IPPP. In Figure 4.14, a bar graph for SNR and RMSE values for various test videos is shown. In Figure 4.14a we observe that, PT-RG-RG2B better than PT-GB-2RGB in terms of SNR for all test videos except for rotation00. In terms of SNR, PT-RG-RG2B performs as good as PT-BSS-IPPP. In terms of RMSE, PT-RG-RG2B outperforms PT-BSS-IPPP. Large improvement is observed in translational motion videos. In case of translation02 videos, at some time instant motion peak dominated HR peak causing larger deviation from RHRT. This problem can be handled by a post-processing algorithm. Thus, it can be conclude that PT-RG-RG2B performs better than PT-GB-2RGB and PT-BSS-IPPP. It can be observed that PT-RG-RG2B performs better than PT-RG-RG2B in terms of SNR though the RMSE is nearly same.

4.7 Conclusion

In this chapter we saw two important colour models, viz. HSI and YCbCr, for improving the extraction of the HR from a video. From experiments we observed that YCbCr performs better than HSI. During analysis of YCbCr, which is not an absolute colour model but a way of encoding RGB information, we observed that simple mathematical operation on RGB components can separate intensity variations from the BVP signal. Analysing RGB colour space with the knowledge of absorptivity of a light by the skin and the sensitivity of camera sensors for different wavelengths of a light revealed two possible combination (PT-RG-RG2B and PT-GB-2RGB) of operation on RGB to give better quality signal. In the last section we compared these two approaches and concluded that PT-RG-RG2B is the strong candidate for further development.
CHAPTER 5

Difference method

In the previous chapter, different colour models are analysed and the chapter is concluded with a candidate algorithm using RGB colour model viz. PT-RG-RG2B. In this chapter, a new technique to improve the performance of the candidate algorithm is discussed. In this technique, improvements in the way RGB traces are extracted, from a video, are suggested (block2 of flowchart in Figure 2.1).

5.1 Average of the difference

When a person in a video is moving, motion between two successive frames is very small (in our case video is captured at 20fps). Thus, majority of patches tracked from one frame to the successive frame do not see significant change in specular reflection component of the received light. Ideally, the only change that can be caused is due to the variation in the diffused light, the part of the incident light which interacts with a vasculature in the skin. If we take the difference between the average RGB values of the successive frames, we get a differentiated RGB traces. The differentiated trace captures small changes in successive frames, which corresponds to the BVP. Integrating such a differentiated traces gives us RGB traces which are expected to be much more cleaner (higher SNR) than the traces obtained by averaging (the way RGB traces are obtained in PT-RG-RG2B).

The patch tracking system is implemented in C++, similar to PT-RG-RG2B. Let $X_R(m,n)$, $X_G(m,n)$ and $X_B(m,n)$ be the summation of red, green and blue pixels in $n^{th}$ patch on $m^{th}$ frame. Let there be $N$ patches which are tracked from frame $m - 1$ to $m$. Then the differentiated traces, $\text{diff}_R$, $\text{diff}_G$ and $\text{diff}_B$ can be given as follows:

$$
\text{diff}_R(m) = \frac{\sum_{n=2}^{N} X_R(m, n) - \sum_{n=1}^{N} X_R(m - 1, n)}{N \times \text{patchSize} \times \text{patchSize}}
$$

$$
\text{diff}_G(m) = \frac{\sum_{n=2}^{N} X_G(m, n) - \sum_{n=1}^{N} X_G(m - 1, n)}{N \times \text{patchSize} \times \text{patchSize}}
$$

$$
\text{diff}_B(m) = \frac{\sum_{n=2}^{N} X_B(m, n) - \sum_{n=1}^{N} X_B(m - 1, n)}{N \times \text{patchSize} \times \text{patchSize}}
$$

(5.1)

Where $\text{patchSize}$ is the length of a patch (Section 3.6). From the differentiated traces, RGB traces are obtained by cumulative summation. Let the RGB traces obtained by cumulative summation be called R, G and B. As the differentiation operation removes the DC component, the normalisation
5.2 Results

In this section we see the performance improvement in PT-RG-RG2B with the difference method. In Figure 5.2, the bar graph of SNR and RMSE values, obtained by processing set of test videos, are plotted. For reference, results of PT-BSS-IPPP are also plotted. In Figure 5.2a we can see that PT-RG-RG2B-AD outperforms PT-BSS-IPPP in terms of SNR for videos with large motion. In Figure 5.2b, color01,02 and rotation01,03 videos, as we discussed in Section 2.3, have no motion or very limited rotational motion. With such a limited motion, algorithm without using differentiated traces (PT-RG-RG2B) gives as good results as with differentiated traces (PT-RG-RG2B-AD). Thus, no further improvement in PT-RG-RG2B-AD is noticed in comparison with PT-BSS-IPPP for video rotation00 and for all the videos in the set translation.

Considering all the merits, we select PT-RG-RG2B as the final algorithm. In the same algorithms, replacing patch tracking by skin pixel tracking gives us SM-RG-RG2B-AD. The results for SM-RG-RG2B-AD shows that it performs better than the PT-BSS-IPPP for videos with no motion or limited motion.
5.3 Conclusion

In this chapter, we have introduced improvements in the way of extraction of RGB traces from a video. These improvements, implemented in PT-RG-RG2B-AD, has given results as good as PT-BSS-IPPP for videos having less motion. For videos having large motion, PT-RG-RG2B-AD outperforms PT-BSS-IPPP by a large margin. Investigating SM-RG-RG2B-AD illustrated that the signal processing by ‘RG-RG2B-AD’ way gives better results than BSS processing.

Figure 5.2: Performance comparison between PT-RG-RG2B-AD and SM-RG-RG2B-AD.
Summary

The aim of our study is to achieve a motion robust algorithm for HR extraction from video analysis. The existing algorithm, referred as FD-BSS, is studied in detail for the motion robustness it claims. We rectified some of the basic ambiguities present in the pre- and post-processing of this algorithm. An attempt to improve the motion robustness with patch tracking is also discussed. In spite of the modification to improve the motion robustness, tests with various videos of a subject with varying degree of motion proved that FD-BSS method gives good results provided that the motion of a subject is limited. With large motion, however, extracted RGB traces are affected by number of signals which is greater than the available mixture signal. The ICA is unable to separate the underlying independent components in the mixture signal if the number of independent signals is larger than the number of mixture signal. Thus, we conclude that use of the ICA is not robust is not efficient with large motion. Moreover, the ICA being a statistical method, requires large Window size. Large Window size has large initial delay to get an output and less sensitivity to the HR variability.

After various attempts to improve FD-BSS, a need for new signal processing algorithm was realised. First, HSI traces were extracted and tested for the BVP signal. The hue component has a strong BVP signal, however, it is affected by large motion of a subject. Next, the experiment with YCbCr revealed that the Cb and Cr components are in anti-phase and adding them together gives better signal. Taking inspiration from YCbCr and analysis of RGB model, a new orthogonal vector pair from RGB components is constructed, viz. R-G and R+G-2B. As these vectors are in a plane orthogonal to the intensity axis, effect of illumination changes component is minimized. The traces extracted from these vectors have same major variation due to illumination changes while slightly different strengths of the BVP. Further improvement in the quality of a signal is achieved by extracting the differentiated traces from a video. A small motion in successive frames of a video, enables to reduce the specular reflection component on the RGB traces by taking the difference value from adjacent frames.

In Figure 6.1 a comparison on the performance metrics is illustrated for various algorithms we discussed in the previous chapters is shown. Average SNR and average RMSE is calculated over different groups of videos with similar motion viz. color, translation and rotation. Primary comparison is between FD-BSS-IPIPP and PT-RG-RG2B-AD. From Figure 6.1a it is clear that SNR is improved. Large improvement with colour and translation video group is observed as compared to rotation. Significant improvement is visible in RMSE in Figure 6.1b. Improvement in colour video group is not significant, but the improvement in translation and rotation groups is by large amount.

In PT-BSS-IPPP, use of ICA involves various matrix operations, such as matrix multiplication, calculating eigenvalue values for matrices etc. In PT-RG-RG2B-AD mathematical operations are very
6.1 Future work

(a) Average SNR computed over video groups.  
(b) Average RMSE computed over video groups.

Figure 6.1: Quantifying the algorithms performance with different type of motion.

simple. Though we did not study the complexity of an individual algorithm, to get an idea about the improvement in term of processing time, we define two parameters. Trace Extraction Time per Frame (TETF), as the name suggests, is the total time required to extract the raw traces from a video divided by the number of frames in a video. In our study, face detection, skin pixel detection or patch tracking, these ROI selection and tracking methods are implemented in C++ using OpenCV. Using time function, the time to extract the raw traces from a video is obtained. The Figure 6.2b shows the TETF for different ROI selection and tracking methods used. Furthermore, the signal processing code implemented in Matlab uses ICA, \( \{ R - G, R + G - 2B \} \) vector pair or \( \{ R - G, R + G - 2B \} \) with average difference. For the HR extraction algorithm, window shifting time can be changed depending on the resolution of the EHRT. Such a change effects the processing time to extract the EHRT from raw traces. Thus, the parameter, Processing Time Per Estimated heart rate (PTPE), is defined which is the total time required to extract the EHRT divided by the number of EHRs in EHRT. The Figure 6.2b illustrates the considerable decrease in the processing time with the algorithm development. Considering the performance of PT-BSS-IPPP and SM-RG-RG2B-AD in Figure 6.1 and processing time in Fig. 6.2, it can be concluded that SM-RG-RG2B-AD performs as good as PT-BSS-IPPP with comparatively very low processing time. Thus, SM-RG-RG2B-AD can be a computationally cheap replacement for PT-BSS-IPPP.

6.1 Future work

- Investigation on use of non-orthogonal axes : In Figure 4.11a, we saw that the vector G-B gave good SNR, however, its orthogonal pair, -2R+G+B in Figure 4.12a,gave poorer SNR. In the investigation of other vectors, it was found that the vectors which are near perpendicular to the skin pixel range yielded better results. The vector R-2G+B also found to give good SNR. The vectors and mapping of HS variation trace with BVP change is illustrated in Figure 6.3. The use of non-orthogonal vector pair to extract better BVP can be investigated in future.

- Better post processing : In this report, while developing the HR extraction algorithm, we mainly concentrated on extracting better raw traces from a video and developing a better pre-processing technique. With large motion, strong changes in intensity and colour, mis-prediction of an EHR increase. The problems caused by the post-processing in FD-BSS are discussed in Section 2.5.3.
6.1 Future work

(a) TETF for used ROI selection and tracking techniques.

(b) PTPE for used signal processing strategies.

Figure 6.2: Computation time comparison.

A better post-processing technique can be developed which takes into account the history along with the SNR of the estimates in the history.

• The group delay: The group delay of a filter is a measure of the average delay of the filter as a function of the frequency. It is the negative first derivative of the phase response of the filter. The band pass filter used in the HR extraction algorithm has a certain delay. In our calculation we did not consider the group delay. The effect of group delay in the algorithm is that the EHRT is delayed by some finite time (depends on the length of the filter) in comparison with RHRT.

• Real time implementation: In the HR extraction algorithm from video analysis, ROI selection and tracking is the most complex and time consuming part. For real time execution of the algorithm, TETF must be lower than $fps^{-1}$. In our case with $fps = 20$, the TETF must be less than 0.05 sec. The graph in Figure 6.2a shows that TETF is greater than 0.05 sec for both the algorithms. For real time implementation of PT-RG-RG2B-AD algorithm a lot of refinement in its implementation would be needed.
6.2 Conclusion

The use of ICA for extracting BVP signal is effective under limited motion of a subject. With large motion there are more underlying signals than the ICA can separate, resulting in poor results. A simple mathematics can be used to extract the BVP signal from a colour video of a subject. Mapping of a colour changes on a plane orthogonal to the intensity axis helps separate the BVP signal from intensity variations. The use patch tracking and skin-map detection for improving motion robustness can be selected depending on the motion robustness requirement. Use of skin-map detection gives simple computationally efficient implementation with reduced capacity to handle motion while patch tracking gives better results with larger motion.
Acronyms

**bpm** beats per minute. 1, 9, 12, 13, 28, 29, 47, 48

**BSS** Blind Source Separation. 5, 6, 40

**BVP** Blood Volume Pulse. 1, 6, 8, 10, 22, 24, 25, 27–32, 34–38, 41, 42, 44

**DHB** De-Oxyhemoglobin. 32

**EHR** Estimated Heart Rate. 6, 10, 12, 13, 42

**EHRT** Estimated Heart Rate Trace. 10, 13, 42, 43, 47, 48

**FD-BSS** Face Detection-Blind Source Separation. 6, 7, 9, 11–15, 21, 22, 25, 41, 42

**FD-BSS-IPPP** Face Detection-Blind Source Separation with Improved Pre and Post Processing. 13–15, 17, 21–24, 41

**FD-BSS-WoPP** Face Detection-Blind Source Separation Without Post Processing. 13, 14

**Hb** Hemoglobin. 31, 32

**HR** Heart Rate. 1, 2, 5–10, 12–15, 22, 24, 25, 27–29, 32, 34, 35, 37, 41–43, 46, 47

**HRT** Heart Rate Trace. 47

**ICA** Independent Component Analysis. 6, 7, 9, 12, 13, 21, 22, 24, 25, 28, 41, 42, 44

**NFFT** N point FFT. 48

**OHb** Oxyhemoglobin. 32

**PPG** Photo-plethysmography. 2–5, 7, 32, 33

**PT-BSS-IPPP** Patch Tracking-Blind Source Separation with Improved Pre and Post Processing. 21–24, 28, 31, 32, 37, 39–42

**PT-BSS-IPPP-IT** Patch Tracking-Blind Source Separation with Improved Pre and Post Processing for Independent Traces. 21–24

**PT-Cb** Patch Tracking - Cb. 29, 31, 32

**PT-Cb-Cr** Patch Tracking - Cb Vs Cr. 29, 31, 32, 35, 37
PT-GB-2RGB  Patch Tracking - GB Vs 2RGB. 35, 37

PT-H  Patch Tracking - Hue. 28, 29

PT-RG-RG2B  Patch Tracking - RG Vs RG2B. 35, 37–39

PT-RG-RG2B-AD  Patch Tracking - RG Vs RG2B with Average Difference. 39–41, 43, 47

PTPE  Processing Time Per Estimated heart rate. 42, 43

RHRT  Reference Heart Rate Trace. 10, 13, 31, 37, 43, 47, 48

RMSE  Root Mean Square Error. 10–14, 22–24, 28, 31, 32, 37, 39–42, 47

ROI  Region of Interest. 6–8, 11, 12, 14–16, 21, 24, 25, 27, 29, 42, 43

SM-BSS-IPPP  Skin Map - Blind source separation with Improved Pre and Post Processing. 17, 22, 23

SM-RG-RG2B-AD  Skin Map - RG Vs RG2B with Average Difference. 39, 40, 42

SNR  Signal to Noise Ratio. 10–14, 22–24, 28, 31, 32, 34, 35, 37, 39–43, 47, 48

TETF  Trace Extraction Time per Frame. 42, 43

traceVis  trace visualisation. 20, 21
Independent component analysis (ICA)

In the study of video analysis for HR, the underlying source signal of interest is the cardiovascular pulse wave that propagates throughout the body. The volumetric changes in the facial blood vessels during the cardiac cycle modify the path length of the incident ambient light such that the subsequent changes in amount of reflected light indicate the timing of cardiovascular events. By recording a video of the facial region with a camcorder, the RGB colour sensors pick up a mixture of the reflected plethysmographic signal along with other sources of fluctuations in light due to artefacts such as motion and changes in ambient lighting conditions. Given that haemoglobin absorptivity differs across the visible and near-infrared spectral range [31], each colour sensor records a mixture of the original source signals with slightly different weights. These observed signals from the red, green and blue colour sensors are denoted by \( x_1(t) \), \( x_2(t) \) and \( x_3(t) \) respectively, which are amplitudes of the recorded signals at time instant \( t \). In conventional ICA the number of recoverable sources cannot exceed the number of observations, thus we assumed three underlying source signals, represented by \( s_1(t) \), \( s_2(t) \) and \( s_3(t) \).

The ICA model assumes that the observed signals are linear mixtures of the sources, i.e. \( x_i(t) = \sum_{j=1}^{3} a_{ij} s_j(t) \) for each \( i=1,2,3 \). This can be represented compactly by the mixing equation,

\[
x(t) = As(t) \quad (A.1)
\]

where the column vectors \( x(t) = [x_1(t)x_2(t)x_3(t)]^T \), \( s(t) = [s_1(t)s_2(t)s_3(t)]^T \) and the square \( 3 \times 3 \) matrix \( A \) contains the mixture coefficients \( a_{ij} \).

The aim of ICA is to find a separating or demixing matrix \( W \) that is an approximation of the inverse of the original mixing matrix \( A \) whose output

\[
\hat{s}(t) = Wx(t) \quad (A.2)
\]

is an estimate of the vector \( s(t) \) containing the underlying source signals. According to the central limit theorem [22], a sum of independent random variables is more Gaussian than the original variables. Thus, to uncover the independent sources, \( W \) must maximize the non-Gaussianity of each source. In practise, iterative methods are used to maximize or minimize a given cost function that measures non-Gaussianity such as kurtosis, negentropy or mutual information [17].

Though ICA is a good mathematical tool there are two ambiguities we should consider. First, the variances (energies) of the independent components can not be determined and Second, the order of the independent components can not be determined [17] i.e. given a set of three signals in a sequence, after ICA, the sequence of the separated signals can not be predicted.
Quantifying the algorithm performance

It is very important to quantify the performance of an algorithm so that its development can be tracked or it can be compared with other algorithms. After executing the HR extraction from video analysis algorithm on a video sequence, a spectrogram is produced. Spectrogram is formed by stacking the power spectrum obtained after each WindowOverlap shift (Section 2.2). Spectrogram obtained by PT-RG-RG2B-AD algorithm on color02 and rotation02 are illustrated in Figure B.1b and B.1c. In ideal case it is expected that a line connecting peaks in a spectrogram, which hereafter called Heart Rate Trace (HRT), tracks the persons HR changes as recorded in RHRT along the time. The HRT extracted from spectrogram gives HR in hertz(Hz). It is multiplied by 60 to give an EHRT in bpm. The important parameter regarding quantifying an algorithm is the error between EHRT and RHRT. Assuming EHRT and RHRT has $n$ samples (depends on the size of a video, Window and WindowOverlap (section 2.1), the RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (RHRT(i) - EHRT(i))^2}{n}}$$ (B.1)

Other important parameter is the cleanliness of a spectrogram. The HRT visible in spectrogram can be strong with strong primary component corresponding to HR and relatively weak second and third harmonic depending on the level of noise. The noise in a spectrogram can have various sources such as illumination changes, colour variation, motion of a person etc. Ideally, we expect a clean spectrogram with single highly distinguishable HRT and few harmonics. Thus, we define another important parameter i.e. SNR. We define SNR of a spectrogram as the average of SNRs of each spectrum in a spectrogram.

The SNR for a spectrum can be calculated as : $SNR = \frac{S}{T}$, where S is the addition of spectral components contributing to the signal and T is the total energy in a spectrum obtained by summing all the spectral components. In our case calculating T is simple but calculating S is ambiguous. In general, if SNR is to be calculated from a spectrum, we should know what the signal is. Once we know the signal in a spectrum, we can sum the spectral components for a signal together to get S. One of the choice for selecting signal is to select it from the HRT. The HRT connects all the peaks in a spectrogram. However, in case of videos with heavy motions, occasionally the peaks due to motion in a spectrum dominates the peak due to the HR signal. Selecting a peak in a spectrum as a signal can occasionally result in selecting a peak due to motion of a subject. In reality, we want to quantify this peak as a noise in our SNR calculation. Thus, using HRT as a signal to calculate SNR can give misleading results.
Other option to get the signal (S) is to consider the RHRT as a signal reference and calculate the S from spectrum. Generally, the EHRT is close to the RHRT but not identical. Also, we allow the EHRT to deviate from the RHRT as long as deviation is by small amount (±5 bpm). Thus, we select RHRT as a signal. In general, if there is a peak, an area marked by a value 50% or 90% of the peak value on either side of the peak is considered as the main lobe of energy across the peak value. It has been observed that the main lobe is often flanked by the peak due to motion. In such a case, the S calculation considers the part of the motion peaks. To avoid this, we define a fixed range of ±10 bpm across the RHRT to calculate S (Figure B.1a). Accumulating all spectral components in this range gives us S and accumulating all the spectral components gives us T. A mathematical definition for calculating SNR of the spectrum is as follows:

\[
SNR = \frac{\sum_{n=A}^{B} spect(n)}{\sum_{n=0}^{NFFT/2+1} spect(n)} - \sum_{n=A}^{B} spect(n) \tag{B.2}
\]

Where, \(A = \left\lceil \frac{(RHR-10) \times NFFT}{fps \times 60} \right\rceil\), \(B = \left\lceil \frac{(RHR+10) \times NFFT}{fps \times 60} \right\rceil\), N point FFT (NFFT) and fps is frame per second for a video. Calculating SNR for each spectrum in a spectrogram gives us a SNR trace as shown in Figure B.1d and B.1e. The average of SNR trace is considered as the SNR for a spectrogram.
Bibliography


[5] Cardoso, J. Matlab implementation of the viola-jones face detection method. http://www.noulo.net/index.php?option=com_content&view=article&id=37&Itemid=60&af9448e7c3876ad3024976acabe1be11=73aea7c56bfc4e6b64b5d78f44d0ee5d.


