Automated non-contact heart rate measurement using conventional video cameras

Gustavo Sandri, Ricardo Queiroz and Eduardo Peixoto

***Abstract*— In this work we propose an algorithm for non-contact heart rate estimation using conventional cameras. The algorithm is robust to movements and works under non-controlled illumination. We divide the frames in micro-regions that are tracked using an optical flow algorithm to compensate for movements and we apply a clustering algorithm to automatically select the best micro-regions to use for heart rate estimation. We also propose a temporal and spatial filtering scheme to reduce noise and an adaptive filter to improve the signal to noise ratio of the signal employed for heart rate estimation. We evaluated our algorithm for real and synthetic data, comparing its results to a fingertip pulse oximeter and a state-of-the-art algorithm for heart rate estimation through video. Results show that the algorithm can work well under challenging situations.**

***Keywords*— Photoplestimography, Video processing, Pulse measurements, Heart rate.**

I. Introduction

Heart rate (HR) is an important measure to ascertain a person's health state. It is traditionally estimated using electrodes or optical sensors that require skin contact and may be uncomfortable. On the other hand, it has been shown that the HR can be estimated using conventional cameras [1], [2], [3], that capture the subtle changes in skin tone, called Photoplethysmographic (PPG) signal. The wide availability of conventional cameras and the ability to contactless estimate the HR have many potential applications.

The signal captured by the camera is a wave that represents the changes in skin tone along time, called the Photoplethysmographic (PPG) signal. The frequency of this wave is the same as the frequency by which the heart beats. Therefore, the PPG signal captured by the camera can be used to estimate a person's heart rate. This remote measurement of cardiac pulse provides more comfort as it avoids the use of electrodes or others devices attached to the body.

Most algorithms use a cascade classifier to detect the subject's face [2], [3], [4], [5] as they have shown that the PPG signal is relatively easy to detect on the face skin. Capdevila *et al.* [6] have shown that the forehead, cheeks and chin are the best regions on the face to measure the PPG signal. They define the region of interest (ROI) as a rectangle comprising most of the subjects face.

Li *et al.* [1] employs Discriminative Response Mat Fitting (DRMF) [7], a more robust face fitting algorithm and define the ROI as the region comprising mainly the cheeks.

The HR is estimated either using only the average value of the green channel inside the Region of Interest (ROI) [1], [8] or a mixture of the average for the red, green and blue channels by means of Independent Component Analysis (ICA) [2], [5], [9] or with fixed weights [10], linearly mixing them in an attempt to maximize the Signal to Noise Ratio (SNR) for the PPG signal. ICA is the preferred approach among researchers.

The signal is converted to the frequency domain using Discrete Fourier Transform (DFT) [2], [3], [4], [5], [11], Short Time Fourier Transform (STFT) [9] or Welch periodogram [1], [12] and the frequency corresponding to the peak of maximum power is attributed as the HR frequency. Some researches also use an approach in the time domain, looking for the position of the peaks and valleys [8]. However, this approach degrades rapidly with noise.

Although these algorithms are able to detect the heart rate for steady videos, they show unreliable estimations for harder situations, such as movements and noise. In this work, we propose an algorithm for HR estimation targeting videos with movements such as people talking and gesturing, which makes it considerably harder to find the PPG signal.

II. Framework

Our framework is composed of six main steps (see Fig. 1).

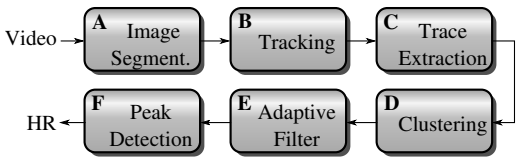


Fig. 1. Schematic of the proposed framework

The video is divided in blocks of 10 seconds to facilitate the tracking algorithm. In **A**, the first frame of the video block is segmented in micro-regions of small size. With a tracking algorithm in **B** we estimate how the micro-regions evolved with time and compensate the movements. We compute the average of the red, green and blue components for the pixels within each micro-region in **C** for every frame. These traces undergo a clustering algorithm in **D** that compare the signal obtained in each micro-region and decide which micro-regions will be used for HR estimation. In this fashion, our algorithm automatically defines the ROI. The traces for the selected micro-regions are combined and converted to frequency domain using DFT. We then apply an adaptive filter to boost its SNR in **E** and then we search for the peak of highest energy and attribute its frequency to the HR in **F**.

A. Image segmentation in micro-regions

The goal of this step is to build a set of micro-regions that will be used to look for the PPG signal. The first frame of each block is segmented in micro-regions by means of the

School of Technology, University of Brasília (UnB), Brasília-DF, Brazil, E-mails: {gustavo.sandri, queiroz, eduardopeixoto}@ieee.org

watershed method [13]. As a pre-processing step to avoid over-segmentation, we apply a bilateral filter [14] (with $\sigma_d = 3$ and $\sigma_r = 0.06$). Then, we convert the frame to gray scale and compute its squared gradient using the Sobel-Feldman operators. The gradient is then further smoothed using a rectangular kernel of size 11x11 and we apply the watershed method. The resulting segments are called micro-regions.

In this step a skin detection algorithm is used in order to eliminate those micro-regions that do not contain at least 80% of skin pixels. The skin detector employ an histogram based approach, as described by Vezhnevets *et. al.* [15], using the database from Jones and Rehg [16] for training.

As skin detection is a hard task that is susceptible to many false positives we also employed the Viola-Jones face detector algorithm [17]. Those pixels that do not fall in a rectangle containing 100% of the height and 80% of the width of the box found by the face detector are set as not being skin pixels.

B. Micro-region tracking

For each micro-region we select a set with up to 12 easy to track points using the algorithm of Shi and Tomasi [18] as implemented in OpenCV 2.4. The tracking of these points is executed using the Lucas-Kanade algorithm as implemented by Yves [19] and is used to compensate for movements.

A sequence of 8 frames is employed for point tracking (Fig. 2). R_0 and R_1 are the reference frames and we know the position of the tracking points for these frames (initially, R_1 is the first frame of the block and the tracking point position at R_0 are found by Lucas-Kanade algorithm). We them apply the Lucas-Kanade algorithm as shown in Fig. 2

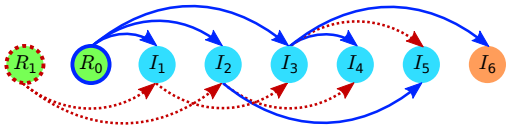


Fig. 2. Optical algorithm: The origin of the arrow indicates the origin frame. The dashed arrows indicates that the optical flow was computed using information coming from reference R_1, and the fill arrows from R_0.

We suppose that between frames R_0 and I_6 the movement of the tracking points can be described by a third order polynomial. The parameters of the movement (initial position, velocity, acceleration and jerk) are found as the ones minimizing the position's squared error. With these parameters we compute the position of tracking points for frames I_1 to I_5. In this fashion, we obtain the temporal filtered position of the tracking points. We then advance in the video sequence. Frames I_4 and I_5 becomes now the reference. We keep advancing on the video sequence until all the video is covered.

From the position of the tracking points at a given time t we estimate an affine transformation that convert the position of the tracking points on the first frame of the block to that at instant t. We then select some key points that represent the micro-region border, as shown in Fig. 3. These points undergo the affine transformation to define how the micro-region evolved with time. The pixels that fall within the delimited region belong to the given micro-region.

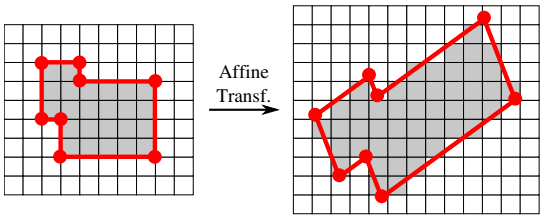


Fig. 3. Key points, that represent the micro-region border, undergo the affine transformation to define how the micro-region evolved with time

C. Trace extraction

For all frames, we compute the average value of the red, green and blue components of all pixels within a micro-region, resulting in $r_i(t)$, $g_i(t)$ and $b_i(t)$, the red, green and blue traces for the i-th micro-region. These traces are combined in a single trace by taking the weighted average of them.

Most algorithms in the literature employ Independent Component Analysis (ICA) to adaptively determine the weights for each channel. However, we observed that the weights attributed to each channel through ICA do not change significantly from one instant to another, or even from one video to another, as shown in Fig. 4.

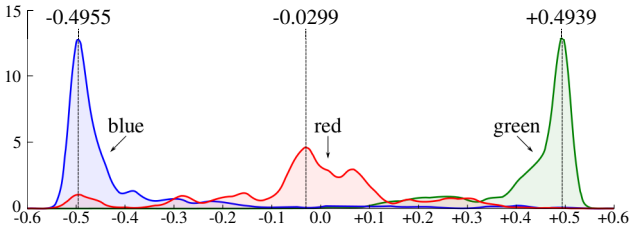


Fig. 4. Estimated probability distribution of the weights attributed to each channel found with ICA for videos of 15 volunteers with a duration of 60 seconds.

In certain cases, the use of ICA may confound the algorithm instead of helping it (this will be discussed in Sec. III). Therefore, we employ fixed weights to mix each channel. The weights were chosen as the values that provide the maximum probability density, as shown in Fig. 4.

D. Clustering algorithm

The traces extracted for each micro-region may contain different levels of noise, depending on parameters such as vasculature, makeup, motion noise, etc.

To obtain a good SNR we try to ignore those micro-regions where the PPG signal is not visible or where the noise energy obscures it. The clustering algorithm is used to resolve which micro-regions to use in order to maximize the SNR. It groups in a cluster those micro-regions that present similar traces comparing their DFT. Based on the assumption that most micro-regions contain the PPG signal, we select the cluster of micro-regions with the highest number of elements and ignore the remaining. In this fashion, the algorithm automatically selects the ROI.

In order to compare the DFT of the traces for each micro-regions we use the distance metric given by

$$D\{F_i, F_j\} = 2\frac{\sqrt{V_{ii}V_{jj}} - V_{ij}}{A_i A_j W_{ij}}, \quad (1)$$

where F_i and F_j are the DFT at the i-th and j-th micro-region,

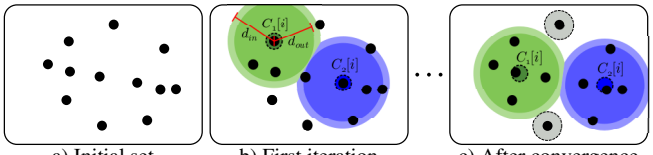
$$V_{ab} = \sum_{v \in \Theta} w_{ij}[v] F_a[v] F_b[v], \quad W_{ij} = \sum_{v \in \Theta} w_{ij}[v],$$

$$w_{ij}[v] = max\left(\frac{F_i[v]}{A_i}, \frac{F_j[v]}{A_j}\right),$$

v is the frequency and Θ is the range of frequencies where we expect the HR to be, that goes from 30 to 240 beats per minute (BPM). A_i and A_j are the root mean squared amplitude of F_i and F_j for $v \in \Theta$. This distance metric is based on the Euclidian distance with the difference that $D\{\alpha F_j, \beta F_i\} = D\{F_i, F_j\}\ \forall\ \alpha$ and β positive real values.

The clustering algorithm proposed in this work is a modification of K-means [20]. One disadvantage of the K-means for our purpose is that we need to know, prior to the clustering, how many clusters we want to form, but the optimal number of clusters is video dependent.

The proposed algorithm is represented in Fig. 5. Its input is $\Upsilon = \{F_n[i]\}$, a set of vectors of the Fourier transform for each micro-region. From Υ, 20% of vectors are randomly selected as cluster center. For the remaining points we calculate its distance to $C_k[i]$ (the cluster center). If this distance is $\leq d_{in} = -2\ ln(0.4)$, the point is integrated to the cluster. Otherwise, if there are no cluster for which the distance is less or equal to d_{in}, the element remains unclassified.



a) Initial set b) First iteration c) After convergence

Fig. 5. Clustering algorithm

Then we update the cluster centers by the average value of the vectors inside it and we compare the cluster centers between them to see how similar they are. Those pairs that have a distance $\leq d_{out} = -2\ ln(0.42)$ are aggregated together. The clusters that remain with a single element are ignored.

Finally, we calculate the distance of all elements within a cluster to its cluster center. Those that have a distance higher than d_{out} are excluded and set as unclassified.

It is possible that after updating the cluster center, some elements within a cluster will actually no longer belong to the cluster. Therefore we calculate the distance of all elements within a cluster to the cluster center. Those that have a distance higher than d_{out} are excluded and set as unclassified.

If we still have unclassified elements, we randomly select 20% of the unclassified elements to form new clusters and we repeat the previous steps till there are no more unclassified elements remaining. At this point, the algorithm is said to have achieved convergence. For some cases it is possible that the algorithm never converges or converge just after a large number of iterations. Hence, we fixed a limit of 200 iterations.

E. Adaptive filter

The traces from the micro-regions selected by the clustering algorithm are averaged in a single trace. Then, the HR is estimated using this trace analyzing it in windows of 30 seconds. We advance in the signal using steps of 0.5 seconds.

The signal inside the window is zero padded to contain a total of 2^{14} elements and we compute the magnitude of its DFT. We retain only those frequencies in the range going from 30 to 240 BPM. An adaptive filtering is then applied over it, multiplying it by a mask $M_n[v]$. The mask aim is to amplify the signal for those frequencies that have a higher probability of be the HR frequency and attenuate others, reducing the effect of noise in the estimation. The mask is defined supposing that the HR varies slowly with time and that within 0.5 seconds it should be almost the same. Therefore, the signals in previous windows provide a good estimation of where the HR peak should be.

Let $Z_n[v]$ be the DFT magnitude at the n-th window, then

$$M_n[v] = (T[v] * Z_{n-1}[v])\,(T[v] * T[v] * Z_{n-2}[v])\,. \qquad (2)$$

$T[v]$ is a triangular function with a support of 2 BPM, centered in 0, that is applied to horizontally stretch the peaks of Z_n in order to accommodate small frequency variations.

Also, it was shown by Xu *et. al.* [21] that the use of the derivative of the traces improves the HR detection performance as the noise tends to be more intense for low frequencies. Hence, we include a high-pass filter to the adaptive filter. The filter employed apply a gain of 0.2 for frequencies inferior to 20 BPM and a gain of 1 for frequencies superior to 150 BPM. Between 20 and 150 BPM it applies a gain that varies linearly with frequency. This filter has a behavior similar to a derivative filter, but we restricted its actuation between 20 and 150 BPM to avoid over-attenuation of low frequencies and over-amplification of high frequencies. We refer to this filter as $F_d[v]$. The filtered signal employed for HR estimation is then $Z_n[v]M_n[v]F_d[v]$.

Finally, we search for the peak of highest amplitude on the filtered signal and attribute its frequency to the HR. If the absolute difference to the previous estimated HR is superior to 12 BPM, we look between the four highest peaks the nearest to the previous estimated HR. If none of them present an absolute difference inferior to 12 BPM we retain the frequency of the highest peak.

III. Results

In order to evaluate the algorithm performance we captured 2 videos of 60 seconds from 35 volunteers in indoors environment with uncontrolled illumination, stored without compression (480x640, 60 fps). In the first video we asked the volunteers to remain as still as possible in front of the camera. In the second video we let the volunteer move freely and we interviewed them to encourage movements. The database is available at [22]. The video from 15 volunteers were employed to define the weights to attribute to each color channel (Fig. 4) and the remaining 20 for HR estimation. Their HR was monitored using a off-the-shelf fingertip pulse oximeter that presents a precision of 2 BPM.

We compared our algorithm to that of Poh *et. al.* citePoh2010,Poh:2011, since most algorithms found

in the literature follow a similar framework than Poh. Fig. 6 presents the percentage of time that the algorithm presented an absolute error, when compared to the pulse oximeter reading, inferior to the given values for the 20 volunteers on the case of steady and videos with movement.

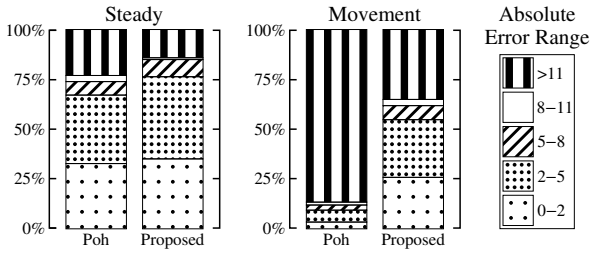


Fig. 6. Algorithm performance to real data. The absolute error range is given in BPM

We can observe that for the steady videos our algorithm presented a slightly better performance. The algorithm of Poh remained 76.94% of the time with an absolute error inferior to 8 BPM, compared to 85.92% for the proposed algorithm. This better performance is due to 2 factors: 1) we avoided the use of ICA to adaptivelly estimate the weights of each color channel. Instead we use a fixed mixture with weights chosen during the training phase; 2) The use of the adaptive filter that boost the SNR of the signal employed for HR estimation.

Since it is hard to evaluate the performance of the algorithm against noise, as we do not know the SNR of the videos in the database, we evaluate the performance against a synthetic trace with known noise, as shown in Fig. 7. This trace is composed of a sine wave with frequency uniformly distributed between 60 and 200 BPM plus a noise signal. This noise is built with a Gaussian noise integrated over time to be more befitting to the noise found in real data that is stronger for lower frequencies. We eliminate the DC component of the noise and multiplied it by a constant in order to obtain the desired SNR. Fig. 7 presents the percentage of time that the proposed algorithm and that of Poh presented an absolute estimated error inferior to 8 BPM for 100 simulations. We also evaluated the performance of the proposed algorithm without the use of the adaptive filter to elucidate its performance.

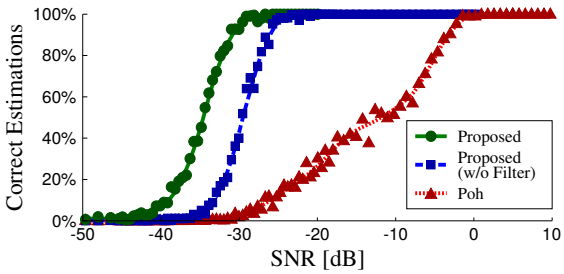


Fig. 7. Algorithm performance to synthetic data

We can observe that for all values of SNR the proposed algorithm presented a higher number of correct estimations than Poh for this simulated noise. Our algorithm achieved 50% of correct estimation at -34.6 dB, while Poh at -11.8 dB. Also, it can be seen that the use of the adaptive filter contributes to a better performance and the proposed algorithm reach the same percentage of correct estimations, in average, 5.2 dB before its version that do not employ the adaptive filter.

The difference in performance is even higher for videos with movement (Fig. 6) because a lot of movement artifacts are introduced in the traces, reducing the SNR. As we compensate the movement in our algorithm we obtain a much better performance with 64.92% estimations with absolute error inferior to 8 BPM, compared to 13.39% for Poh.

Fig. 8 depicts the regions that were chosen most of the time to compose the ROI by the clustering algorithm. For the steady videos, the region of the forehead and the cheeks are preferred, which should be expected as these are the most vacularized regions on the face. On the other hand, for videos with movement, mainly the forehead is chosen to compose the ROI while the cheeks are eliminated most of the time due to movement artifacts.

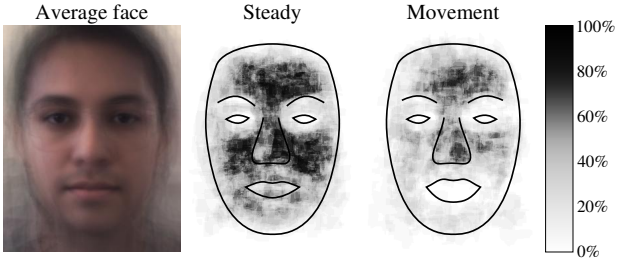


Fig. 8. Percentage of time that a region was chosen for HR estimation. The average face was computed after aligning the volunteers face with respect to the mouth and eyes

IV. Conclusion

In this work we propose an algorithm for heart rate (HR) estimation using videos of the human face under uncontrolled light in indoor environments. We compared our results to that of Poh [2] and observed a substantial improvement.

The algorithm employs an adaptive filter that imposes a temporal coherence on the signal and is based on the assumption that the heart rate varies slowly with time. A derivative filter was also employed to reduce the influence of low frequency noise. These filters boosts the signal to noise ratio of the signal used for HR estimation and we showed that they are capable of reducing the number of incorrect estimated HR.

Also, the motion compensation combined with the clustering algorithm improved the performance, mainly for the videos with movement. The clustering algorithm automatically select the best micro-regions to employ, eliminating noisy ones, which contribute to improve the traces' SNR.

References

[1] X. Li, J. Chen, G. Zhao, and M. Pietikainen, "Remote heart rate measurement from face videos under realistic situations," in *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, June 2014.

[2] M.-Z. Poh, D. J. McDuff, and R. W. Picard, "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation." *Optics express*, vol. 18, no. 10, pp. 10 762 – 10 774, 2010.

[3] T. Pursche, J. Krajewski, and R. Moeller, "Video-based heart rate measurement from human faces," *Electronics (ICCE)*, pp. 544 – 545, 2012.

[4] M.-Z. Poh, D. McDuff, and R. Picard, "Advancements in noncontact, multiparameter physiological measurements using a webcam," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 1, pp. 7–11, January 2011.

[5] S. Kwon, H. Kim, and K. S. Park, "Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2174 – 2177, 2012.

[6] L. Capdevila, J. Moreno, J. Movellan, E. Parrado, and J. Ramos-Castro, "Hrv based health amp and sport markers using video from the face," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, August 2012, pp. 5646 – 5649.

[7] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3444 – 3451.

[8] J.-P. Couderc, S. Kyal, L. Mestha, B. Xu, D. Peterson, X. Xia, and B. Hall, "Pulse harmonic strength of facial video signal for the detection of atrial fibrillation," in *Computing in Cardiology Conference (CinC), 2014*, September 2014, pp. 661 – 664.

[9] Y.-P. Yu, B.-H. Kwan, C.-L. Lim, S.-L. Wong, and P. Raveendran, "Video-based heart rate measurement using short-time fourier transform," in *Intelligent Signal Processing and Communications Systems (ISPACS), 2013 International Symposium on*, November 2013, pp. 704 – 707.

[10] U. Bal, "Non-contact estimation of heart rate and oxygen saturation using ambient light," *Biomed. Opt. Express*, vol. 6, no. 1, pp. 86 – 97, January 2015.

[11] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, "Remote plethysmographic imaging using ambient light," *Opt. Express*, vol. 16, no. 26, pp. 21 434 – 21 445, December 2008.

[12] J. Bolkhovsky, C. Scully, and K. Chon, "Statistical analysis of heart rate and heart rate variability monitoring through the use of smart phone cameras," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, August 2012, pp. 1610 – 1613.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, third edition ed. Prentice Hall, 2007.

[14] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, January 1998, pp. 839 – 846.

[15] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *IN PROC. GRAPHICON-2003*, 2003, pp. 85 – 92.

[16] M. Jones and J. Rehg, "Statistical color models with application to skin detection," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1, 1999, p. 280.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–(511 – 518).

[18] J. Shi and C. Tomasi, "Good features to track," in *Proceedings CVPR 94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593 – 600.

[19] B. J. Yves, "Pyramidal implementation of the lucas-kanade feature tracker," *Microsoft Research Labs, Tech. Rep*, 1999.

[20] K. Sayood, *Introduction to Data Compression*, fourth edition ed., K. Sayood, Ed. Morgan Kaufmann, 2012.

[21] S. Xu, L. Sun, and G. K. Rohde, "Robust efficient estimation of heart rate pulse from video," *Biomed. Opt. Express*, vol. 5, no. 4, pp. 1124 – 1135, April 2014.

[22] [Online]. Available: https://image.unb.br/˜gustavo/HR_Database