Remote PPG with POS and 2SR

Alimzhan Sultangazin
UID: 005035952
UCLA
asultangazin@g.ucla.edu

Jonathan Bunton
UID: 405218799
UCLA
jonathan.michael.bunton@gmail.com

Abstract

This paper explores the task of remote photo-plethysmography (rPPG), where we use standard RGB videos to estimate a human subject's pulse signal. We construct a simple proof-of-concept algorithmic pipeline using both the Spatial Subspace Rotation (2SR) and Plane-Orthogonal-to-Skin (POS) methods, requiring only one initial skin-pixel identification step. We then verify our pipeline's output with a small sample from two data sets, and discuss potential areas for improvement.

1. Introduction

Photo-plethysmography (PPG) is a well established method for extracting human vital signs such as pulse signal, blood oxygen content, and heart rate from light reflections on the skin. Traditionally, extracting these signals requires a precise LED light source that is in contact with the subject's skin, to avoid signal interference.

In this work, however, we tackle the slightly more difficult task of *remote* photo-plethysmography (rPPG). The key difference between remote methods and traditional PPG methods is that we no longer require the subject to be in direct contact with the light source. Instead, rPPG takes a simple color (RGB) video of the subject and extracts the subject's pulse signal by analyzing variations in the subject's skin hue over time.

There are a wide variety of algorithms for rPPG in literature [9, 7, 2, 3, 6, 4], but in this work we leverage and investigate two recently developed methods. In particular, we use the Spatial Subspace Rotation (2SR) and the Plane-Orthogonal-to-Skin (POS) methods, which operate under similar guiding principles. We highlight their similarities and distinctions in the following sections, and compare their effectiveness on some proof-of-concept sample data provided both by the course and by the references in [1].

2. Methods

In this work, we design a processing pipeline that takes a standard RGB video of a human subject, extracts the skin pixels with mild supervision, then estimates the subject's pulse signal with one of two established methods. Finally, we perform frequency analysis to estimate the instantaneous heart rate of the subject. The entire algorithmic pipeline is illustrated in Fig. 1.

2.1. Skin Detection

Given an arbitrary video of a subject, only a small number of recorded video pixels contain the signal we seek– those corresponding to the subject's skin. The natural first step in our pipeline is to extract exactly these pixels from our raw input videos.

Inspired by prior work in [9] and [8], we turn to kernelized one-class support vector machine (OC-SVM) to identify which pixels correspond to the subject's skin. To be precise, we begin by asking the user to identify several small regions of skin pixels in the first frame of the video. We then extract these regions in the first few frames and mark them as positive features for the kernelized OC-SVM.

Once the OC-SVM is trained with these features, we use it to classify pixels as skin or non-skin in each frame of the provided video. The resulting "stripped" video of only skin pixels is fed into the proceeding algorithms to estimate the pulse signal.

2.2. Spatial Subspace Rotation (2SR)

The spatial subspace rotation methodology, as described in [9], is performed in two steps:

1. find the principal components of the skin pixels from each video frame in RGB space;

2. determine the change in magnitude and orientation of the dominant principal component over time for pulse extraction.

Let $N \in \mathbb{Z}_{>0}$ be the number of skin pixels in an image at time t. We construct matrix $I(t) \in \mathbb{R}^{N\times 3}$ by stacking the RGB values of skin pixels. Let the singular value decomposition (SVD) of matrix $I(t) = U(t)\Sigma(t)V^T(t)$, where $\Sigma(t) \in \mathbb{R}^{N\times 3}$, $U(t) \in \mathbb{R}^{N\times N}$, and $V(t) \in \mathbb{R}^{3\times 3}$. The

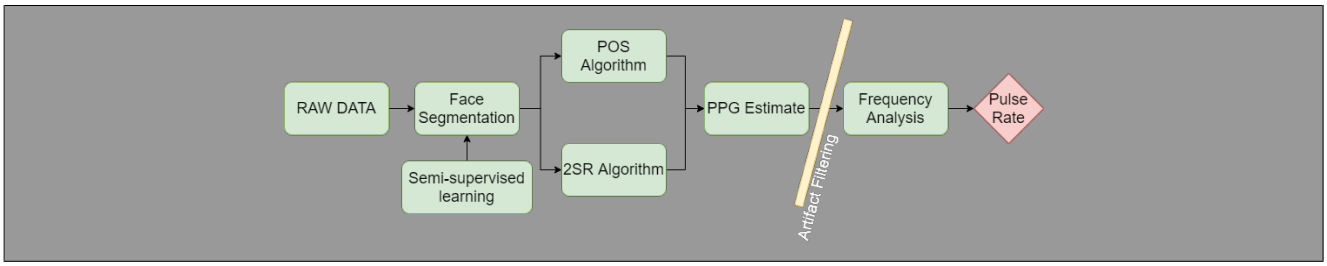

Figure 1. A flowchart illustrating our algorithmic pipeline. Data is input as a sequence of video frames containing a human subject. An external (human) source identifies some regions of skin pixels for the OC-SVM algorithm, which then removes irrelevant pixels from the video frames. The stripped video is then processed by either/both the 2SR or POS algorithms, creating estimates of the pulse signal. This estimated signal is filtered and its spectrum is analyzed via periodogram to estimate the subject's heart rate.

principal components of the skin pixels in RGB space are given by columns of matrix V, while singular values in Σ give a metric of how spread out the skin pixels are in corresponding components (we convert those into eigenvalues by $\lambda_i(t) = \frac{\Sigma_{ii}(t)^2}{N}$).

The change in magnitude and orientation of the dominant principal component $v_1(t)$ is found over a sliding window. Two quantities of interest to us are the rotation between $v_1(t)$ and the plane defined by the next largest principal components $v_2(\tau)$ and $v_3(\tau)$, which we calculate as:

$$R(t,\tau) = (v_1^T(t)v_2(\tau), v_1^T(t)v_3(\tau)), \tag{1}$$

and the scaling of $\lambda_1(t)$ with respect to $\lambda_2(\tau)$ and $\lambda_3(\tau)$:

$$S(t,\tau) = (\sqrt{\lambda_1(t)/\lambda_2(\tau)}, \sqrt{\lambda_1(t)/\lambda_3(\tau)}). \tag{2}$$

To emphasize the effect that blood flow in the subject has on both of these quantities, we element-wise multiply these quantities and backproject:

$$SR^b(t,\tau) = \begin{bmatrix} v_2(\tau) & v_3(\tau) \end{bmatrix} \cdot SR(t,\tau). \tag{3}$$

Let us denote the first and second row of $\overrightarrow{SR}^b(\tau)$ by $\overrightarrow{SR}^b_1(\tau)$ and $\overrightarrow{SR}^b_2(\tau)$, respectively. Before aggregating the data from each window, we preprocess it into a vector $\overrightarrow{SR}^{agg}(\tau)$:

$$\overrightarrow{SR}^{agg}(\tau) = \overrightarrow{SR}^b_1(\tau) - \frac{\sigma(\overrightarrow{SR}^b_1(\tau))}{\sigma(\overrightarrow{SR}^b_2(\tau))}\overrightarrow{SR}^b_2(\tau), \tag{4}$$

where σ is the standard deviation operator. Finally, we aggregate the data for all windows by adding it to the pulse signal $\overrightarrow{p}$ as follows:

$$\overrightarrow{p}^{\tau}(\tau : \tau + l - 1) = \overrightarrow{p}^{\tau-1}(\tau : \tau + l - 1) + \overrightarrow{SR}^{agg}(\tau), \tag{5}$$

where $\overrightarrow{p}^{\tau}$ denotes the values of the pulse signal at τ^{th} iteration of the algorithm. We initialize by $\overrightarrow{p}^0 = 0$.

2.3. Plane-Orthogonal-to-Skin

The second algorithm for pulse signal extraction we consider is the Plane-Orthogonal-to-Skin method introduced in [7]. The algorithm follows the same key steps as 2SR outlined above, namely:

1. identify a representative vector in RGB space defining the skin tone of the subject;

2. track deviations from this representative vector (the pulse signal) by projecting the RGB video signal onto planes orthogonal to the vector.

The key difference in the POS algorithm as opposed to 2SR lies in how the representative skin hue (i.e. a skin vector in RGB space) is computed, and then how this pixel's variation is measured.

In the POS algorithm, we begin by computing the average RGB vector $\mathbf{C}(n) \in \mathbb{R}^3$ of each frame n in the stripped video. We do this to mitigate effects caused by lighting changes, as well as small movements in the subject's face.

We next compute the "reference" representative RGB vector that we will track deviationa from, $\mathbf{C}_{ref} \in \mathbb{R}^3$, by taking an average over a time window of length W, i.e. $\mathbf{C}_{ref} = \frac{1}{W}\sum_{i=n}^{n+W}\mathbf{C}(i)$. The data within this window, $\mathbf{C}(i)$, is then adjusted so that $\mathbf{C}_{ref} = \mathbf{1} \in \mathbb{R}^3$.

For each frame i in the averaging window, we compute the projection of the frame's average RGB vector $\mathbf{C}(n)$ onto planes orthogonal to the skin vector $\mathbf{1} \in \mathbb{R}^3$, which minimizes the impact of skin hue brightness variations from light and movement. Finally, the resulting one-dimensional signals are re-combined with a tuning parameter dependent on the standard deviations, selected to maximize the pulse signal strength.

This process is repeated while sliding the window of length W across the entire stripped video data. This window is typically selected to be approximately 1.6 seconds in length [7]. At each placement of the window, the resulting estimate is overlap added to create a more robust and accurate pulse signal estimate.

2.4. Pulse Signal Filtering

The pulse signal estimates produced by both the 2SR and POS algorithms still contain various artifacts resulting from noisy measurements and compression artifacts.

In our examples–particularly in the Instructor provided data set discussed in Section 3.2–there were video compression artifacts causing a significant oscillation of all video pixels at frequencies of 1 and 2 Hz. To mitigate this, we applied bandstop filters with a width of 0.04 Hz around these regions for this data set.

In addition, human physiology tells us that reasonable heart rates ought to be in the range of 40-200 beats per minute (this is a *very* conservative definition of "reasonable"). These heart rates correspond to a frequency band of around 0.6 - 3.3 Hz, so we apply a bandpass filter over these frequencies to our estimated pulse signal. This bandpass filtering produces smoother and cleaner results from our somewhat noisy estimated signal.

2.5. Heart Rate Estimation

Once we have constructed and appropriately preconditioned an estimate for the pulse signal, we must estimate the subject heart rate. Intuitively, the heart rate should be the most consistent periodic signal in the estimated pulse. If we examine the frequency spectrum of the estimated signal then, the frequencies with the largest contribution (highest power spectral density) ought to be from the subject's heart rate. If we simply compute the power spectral density of the entire estimated pulse signal, the highest peak should correspond to the average heart rate over the interval.

If we are instead interested in the instantaneous heart rate at various times throughout the given video, we could consider computing the power spectral density along windows of our estimated signal. This idea turns out to be precisely the *periodogram* of the signal. Given our estimated pulse signal $\mathbf{H}(i)$, for $i = 1, 2, ..., N$, the periodogram is a set of power spectral densities $\mathbf{P}(i)$ computed at several sampled time windows $i = 1, 2, ..., \tilde{N}$. If we select the maximum frequency component (the highest peak) of the power spectral density $\mathbf{P}(i)$ at each sampled time window i, we recover instantaneous estimates of the heart rate.

The size of this window in our periodogram method represents a trade off between accuracy and time fidelity. A larger window includes more measurements and therefore produces a more accurate estimate of the heart rate. However, a larger window means less time windows can be computed, therefore we lose some resolution in the time domain.

3. Results

We validate our results on two small data sets: a set of three videos of different subjects recorded in [1], and another provided by the course Instructor. We highlight proof-of-concept results on each data set individually using the 2SR and POS algorithms below.

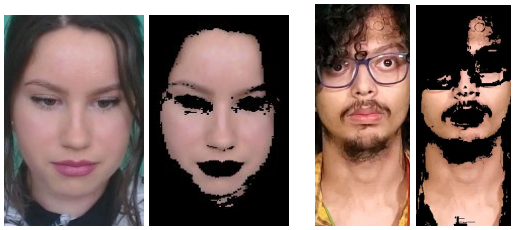

Figure 2. Sample frames before and after skin detection using OC-SVM for a subject in the external data set from [1] (left), and data provided by the course Instructor (right).

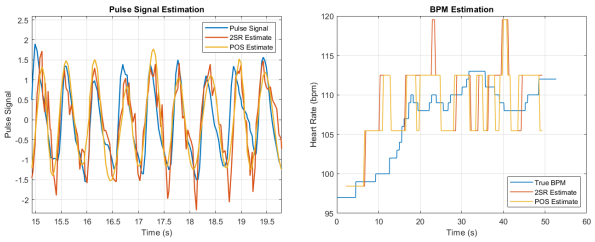

Figure 3. Visual comparison of the true and estimated PPG signal (left) and instantaneous heart rate (right) by both the 2SR and POS algorithms on a sample subject from the external data provided by [1].

3.1. External Data

The small sample of data provided in [1] consists of three different subjects completing a math puzzle while sitting in front of a green backdrop. Simultaneously, the subject's PPG signal and instantaneous heart rate are recorded. Because this data set provides ground truth PPG and heart rate signals, we can compare the accuracy of our PPG signal and heart rate estimation processes separately.

For all videos, the skin detection algorithm was very effective. This accuracy is unsurprising, as each video is stripped using a machine learning model uniquely tailored to the subject in question. For reference, a sample stripped frame from one subject video is shown in Fig. 2.

The actual PPG signal provided with the data set was very close to the estimated signal from both the 2SR and POS algorithms (see Fig. 3 for visual reference). To verify this closeness rigorously, we computed Pearson's correlation coefficient for 2SR and POS's outputs for each subject in Table 1. In all cases, the coefficient is above 0.6, indicating good correlation between the true PPG signal and the estimate.

For heart rate estimation, we see promising results across all subjects. Over large windows, the heart estimate from both 2SR and POS follows the average heart rate from the subject closely. Both methods also showed fair, but less

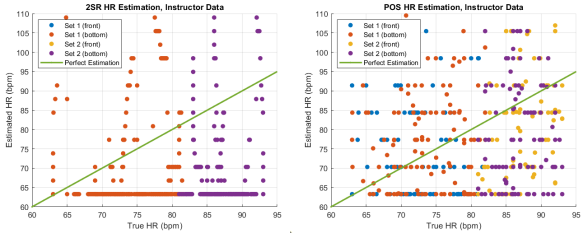

Figure 4. Scatter plots showing the relationship between the instantaneous heart rate estimates by the 2SR algorithm (left) and the POS algorithm (right) for the instructor data sets.

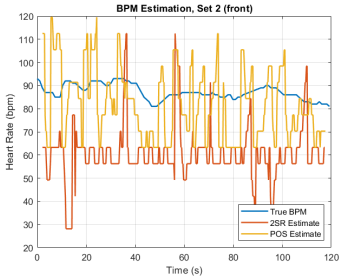

Figure 5. Estimates of the instantaneous heart rates by both the 2SR and POS algorithms compared with the true instantaneous heart rate in one video of the instructor's data set.

impressive results on instantaneous heart rate estimation, as shown in Fig. 3.

3.2. Instructor Data

The data provided by the course Instructor consists of four videos with simultaneous heart rate measurements for each. Because of this format, we can only compare our full algorithmic pipeline outputs with the provided data set. In other words, we must convert the output of 2SR and POS into instantaneous BPM measurements and only then we can compare with the ground truth measurements.

We can see from Table 1 that the BPM signals provided by both algorithms have a low correlation with the ground truth data. Moreover, the parameter for statistical significance (p-value) of this correlation is fairly high (order 0.1) in all cases. From Figure 4, we observe that the majority of BPM estimates are significantly off. Also, note in Figure 5 that most of the estimates provided by the algorithms are at 60 BPM. We conjecture that this phenomenon is caused by compression or lighting artifacts present in video files.

4. Discussion and conclusions

In this work we considered the problem of estimating a photo-plethysmogram (PPG) signal from remote videos of subjects. We have considered two algorithms: spatial subspace rotation (2SR) and Plane-Orthogonal-to-Skin (POS). Both algorithms determine PPG by observing the motion of a "representative" vector in RGB space. The only difference is that 2SR chooses the "representative" vector to be the main principal component of skin pixels, while POS chooses that to be their average. Applying these algorithms on external data from [1] has produced adequate results when compared with ground truth. This, however, was not the case when these algorithms were applied to the data provided by the instructors. Our conjecture is that movement, lighting issues, and compression artifacts may have poorly conditioned these videos for our approach.

Set	2SR	POS
1	0.676	0.64
3	0.685	0.60
4	0.685	0.64

Set	2SR	POS
1 (front)	0.042	0.032
1 (bottom)	0.042	0.034
2 (front)	0.114	0.3167
2 (bottom)	0.114	0.3595

Table 1. Pearson's correlation coefficients (all measured with statistically significant $p < 0.01$) between the true and estimated PPG signal (left) and between the true and estimated instantaneous heart rate (right) for both 2SR and POS.

Implementing these algorithms, we noted that many decisions made by the authors of [9] and [7] were heuristic (e.g., finding the rotation of the main principal component with respect to two other principal components). Therefore, as part of future work, we suggest that the motion of the "representative" vector be modelled using system identification methods from control theory. The idea is to fit the motion of the representative vector for data from a sliding window to a linear dynamical system. The imaginary component of the dominant eigenvector of the system should give us the frequency of pulses in this window.

From a computational imaging point of view, based on the discussion from [7], we think that we might benefit from shining green light on the subject when taking the videos as the green component of light is more sensitive to blood flow influences. Validating this requires us to improve the experimental setup because the preliminary experiments have shown that the current equipment (i.e., Fitbit tracker) does not measure pulse data and measures BPM every 5 seconds only. In addition, the algorithms might benefit from filtering the direct component of light. However, the algorithms separating direct and global components often involve averaging images over video (e.g., [5]) which would destroy the pulse information.

References

[1] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82 – 90, 2019. Award Winning Papers from the 23rd International Conference on Pattern Recognition (ICPR).

[2] G. de Haan and V. Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.

[3] D. G. Haan and van Aj Arno Leest. Improved motion robustness of remote-ppg by using the blood volume pulse signature. *Physiological Measurement*, 35:1913–1926, 2014.

[4] M. Lewandowska, J. Rumiński, T. Kocejko, and J. Nowak. Measuring pulse rate with a webcam — a non-contact method for evaluating cardiac activity. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 405–410, 2011.

[5] Shree K. Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph.*, 25(3):935–944, July 2006.

[6] M. Poh, D. J. McDuff, and R. W. Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE Transactions on Biomedical Engineering*, 58(1):7–11, 2011.

[7] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2017.

[8] W. Wang, S. Stuijk, and G. de Haan. Exploiting spatial redundancy of image sensor for motion robust rppg. *IEEE Transactions on Biomedical Engineering*, 62(2):415–425, 2015.

[9] W. Wang, S. Stuijk, and G. de Haan. A novel algorithm for remote photoplethysmography: Spatial subspace rotation. *IEEE Transactions on Biomedical Engineering*, 63(9):1974–1984, 2016.