Heart Rate Estimation using Spacial Subspace Rotation

Andrew Ho (005629123)
University of California, Los Angeles (UCLA)
aksh98md@g.ucla.edu

Kasra Jalali-Nadoushan (505625981)
University of California, Los Angeles (UCLA)
kasraj@g.ucla.edu

Loic Maxwell (504422999)
University of California, Los Angeles (UCLA)
loicmaxwell17@g.ucla.edu

Mine Gokce Dogan (305430282)
University of California, Los Angeles (UCLA)
minedogan96@g.ucla.edu

Abstract

In this report, we describe an algorithm pipeline for extracting a subject's heart rate using only their frontal-facing video footage. The fundamental component of our pipeline is the Spacial Subspace Rotation (2SR) algorithm, which extracts the subject's PPG signal by tracking the subtle changes in the subject's skin tone over time. Our pipeline performed well on provided datasets, with average errors per data point ranging from 2.9-5.2 beats per minute (BPM) and average errors between data averages ranging from 0.4-2.9 BPM. Furthermore, we evaluated various computational imaging techniques for improving the performance of our pipeline, such as varying the camera's exposure and ISO.

1. Introduction

Remote Photoplethysmography (rPPG) is a process used to extract a subject's heart rate from a video of the subject's face. The most important step in rPPG is to find a successful algorithm for the pulse extraction. There are several robust pulse extraction algorithms in the literature such as CHROM [1], PBV [2] and 2SR [3]. Although all these algorithms share a common spatio-temporal scheme, they use different priors to combine the RGB traces [3]. In particular, CHROM assumes a standardized skin-color and PBV uses a predefined pulse signature [3]. However, these fixed priors can be suboptimal if the relative contribution of blood volume pulse to the RGB channels is changed. Therefore, we decided to use 2SR algorithm in this project since it solves the limitations of conventional rPPG algorithms.

In this paper, we use the following pipeline to measure heart rate of subjects in multiple videos. First we obtain a window containing the face and neck using a face detection algorithm. This window is then used to generate a skin map through a skin detection algorithm that separated the skin pixels from non-skin pixels. This skin map is then used for pulse extraction with 2SR. Using the extracted PPG signal, we then utilize peak detection to help find the inter-beat periods to estimate the heart rate of the subject. Additionally, we evaluated novel computational imaging strategies for improving the accuracy of our algorithm. We first observed how changing the ISO and exposure when capturing the video affects the algorithm's performance. Finally, we researched how the algorithm could be improved using different imaging and machine learning techniques, like spectral filtering and more advanced skin segmentation.

2. Proposed Image Processing Pipeline

2.1. Face Detection

In order to provide an accurate skinmap to the 2SR algorithm, a skin segmentation method must be used on each of the video frames to extract the subject's skin pixels. Running skin segmentation on entire frames results in unwanted artifacts due to background objects, clothing, or hair whose color closely matches that of the subject's skin, since they are miss-classified as skin pixels. Face detection was used to help mitigate the noise due to these artifacts. A pretrained convolutional neural network [4][5] was selected due to its flawless performance on all processed datasets.

The 2SR algorithm generally performs better when provided with a larger set of skin pixels. Once the face was detected, the subject's neck was also included by extending the bottom side of the face's bounding box by an amount that equals the total height of the original bounding box. Although this technique necessarily includes small portions of the background and the subject's shirt since an individual's face:neck length ratio changes from person to person, it adds the benefit of providing significantly more skin pixels to the algorithm.

2.2. Skin Segmentation

After obtaining a window containing the face and neck, we ran a skin detection to get a measurement of skin pixels in a given frame. We used a number of thresholds to accomplish this segmentation. Initially, we obtained a threshold on a gray-scale image of the frame to remove the background. We then obtained thresholds using both the YCbCR and HSV color spaces to increase the robustness of our skin detection algorithm[6]. This is because the RGB color space is more sensitive to different light conditions than the YCbCR and HSV [6]. After obtaining color spaces thresholds for the YCbCR and HSV color spaces, we combined them and passed the background removed image through this combined threshold to obtain the skin map. Figure 1 shows the skin map obtained through our segmentation.

Figure 1: Image before and after thresholding passes

2.3. Pulse Extraction

For the pulse extraction algorithm, we decided to use 2SR [7] because it works effectively regardless of the subjects' state of motion. It is worth noting that we also considered another robust algorithm, POS [8]. We decided to use 2SR in favor of the POS algorithm because we deemed the difference in performance between the two algorithms negligible. Using the previous computed skinmap, a subspace of the provided skin pixels was computed in the RGB color space [3]. Then, the rotation angle of these subspaces between adjacent frames was calculated over the length of the entire video, from which the PPG signal could be extracted. We refer readers to [3] for a more detailed explanation of the algorithm.

2.4. Heart Rate Estimation

2.4.1 Peak Detection

In the heart rate estimation process, the first step is to find interbeat intervals in the PPG signal by detecting systolic peaks. For this purpose, we employed a peak detection algorithm called automatic multiscale-based peak detection (AMPD) [9], [10]. This is an efficient algorithm for peak detection in noisy periodic and quasi-periodic signals. This algorithm is based on analysis of local maxima scalogram that is a matrix comprising the scale-dependent occurrences of local maxima [9]. We decided to utilize this algorithm due to the following three important properties. Firstly, the AMPD algorithm is used to detect peaks in periodic and quasi-periodic signals [9]. Since the dynamic behavior of the PPG signal is mostly quasi-periodic on a small timescale, the AMPD algorithm is suitable for our task. Secondly, the AMPD algorithm does not require us to select free parameter values in advance [9]. Finally, the AMPD algorithm is robust against low and high frequency noise [9]. We refer readers to [9] for a more detailed explanation of the algorithm.

2.4.2 Estimation Process

After detecting the systolic peaks in the PPG signal, we found the interbeat intervals (IBI) to estimate the heart rate. An interbeat interval refers to the time interval between a pair of successive peaks. In order to find the IBI, we first need to find the number of frames between successive peaks. Since the frame rate is 30 frames/second in our dataset, we multiplied the number of frames between successive peaks with the inverse of the frame rate to find the interbeat intervals. After calculating these intervals, the heart rate estimate $HR = \frac{60}{IBI}$.

In peak detection process, algorithms are likely to detect false peaks caused by motion artifacts and other kinds of noise. Therefore, we applied a certain threshold to the peaks. For the provided dataset, we chose this threshold as 120 BPM since the normal resting heart rate is between 60-100 BPM. In general, rPPG methods do not perform well when the person exercises. Therefore, assuming that the person rests during measurement, this threshold generally results in successful estimates. While applying this threshold, we performed the following procedure. If a pair of peaks resulted in an estimate higher than our threshold, we ignored the second peak and estimated the heart rate by using the next peak. This way, we were able to eliminate overestimates. Finally, we applied a moving average filter because there were drastic changes in our estimates. The first three steps of the smoothing procedure are as follows.

$$\begin{aligned} \tilde{y}_1 &= y_1 \\ \tilde{y}_2 &= \frac{y_1 + y_2 + y_3}{3} \\ \tilde{y}_3 &= \frac{y_1 + y_2 + y_3 + y_4 + y_5}{5} \end{aligned} \tag{1}$$

where y keeps the heart rate estimates and $\tilde{y}$ keeps the smooth heart rate estimates.

2.5. Incorporation of Computational Imaging

Computational imaging aggregates information from multiple images and additional dimensions of data captured, such as time, position, or frequency. By targeting specific dimensions that demonstrate a variance throughout the signal, the heart rate estimation accuracy can be increased.

2.5.1 Spectral Filtering

In [11], the authors note that in performing remote PPG tasks, green color and infrared frequency variance is correlated with the heart beat, based on an analysis of the anatomy of the skin. The experiments in the work are conducted on a finger at close range, on both compressed and uncompressed skin. Therefore, it can be applied to video capture at longer distances. A infrared and visible light camera would be used, with a sufficient high resolution to capture the details of the subject's skin. Additionally, a color filter array would be implemented to increase the spectral sensitivity on the specific frequencies that have a higher signal-to-noise ratio (SNR). Further analysis should be done verifying that the PPG signal can be extracted from green and infrared frequencies in plain video, as well as identify additional frequencies that may aid the detection based on SNR.

2.5.2 Multiple Exposure Capturing

One of the shortcomings of the previously proposed approach is the dependency on the accurate segmentation of skin pixels, as described in 2.2, which uses color to differentiate. An alternative method is to use the texture as means of differentiation [12], since skin texture differs from that of clothing or background, so it can be exploited to perform skin segmentation.

The texture captured is highly dependent on the exposure of the photo, or how much light is shone on the subject. To capture the textures effectively, various exposures can be used to capture the same subject, creating multiple images for each frame that have different exposures at the same time point. Each of these images are segmented individually based on the texture, then the intersection of the segmented areas from all images are taken, essentially finding the overlap of skin from low to high exposures. The capturing of different exposures can be done by varying the ISO or the exposure time, which can control how much light is let in to the lens, by means of cycling through a low, medium, and high settings. However, this is a difficult task to perform with an ordinary camera, as the cycle of adjustments to the camera capture mode would need to be made after every image to ensure near-identical pictures are captured, which is need for the segmentation intersection of various exposures.

3. Results and Analysis

3.1. Proposed Image Processing Pipeline

The four given datasets were provided to our algorithm pipeline. In Figure 2, the extracted PPG signals for the provided datasets can be seen. It seems like our pulse extraction method performs as expected since the behavior of our PPG signals is similar to the quasi-periodic signals such as actual PPG signals. The computed heart rate for each dataset was plotted against the ground truth heart rate in Figures 3 and 4.

We can clearly see how the moving average filter has helped smooth out the heart rate computation. The respective errors are tabulated in Table 1.

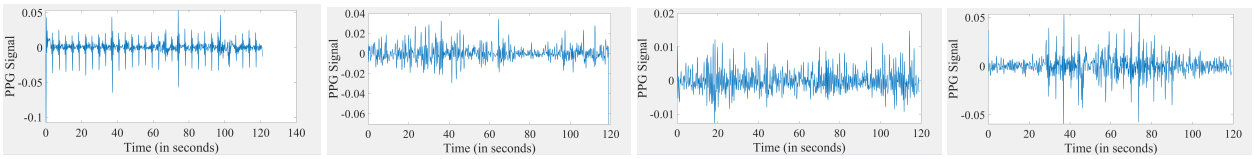
Overall, our heart rate estimations were relatively close to the ground truth, with data point average errors staying between 2.8 and 5.2 BPM. Our pipeline also performs better when computing an average heart rate over many points, as can be seen in the low errors (less than 3 BPM) in the second column. These errors can be attributed to a couple of factors: the skin segmentation and the data alignment procedure. Although our face detection was able to remove most unwanted artifacts, there were still some non-skin pixels from the subject's shirt and hair that were captured during the skin segmentation process. This added noise to the PPG signal that was extracted using 2SR. Additionally, the heart rates that were calculated from the PPG signal were not temporally aligned with the ground truth heart rates. An alignment procedure was carried out for each data point, which consisted of taking a weighted sum of the previous and next closest data points in time. This assumes that the computed heart rate linearly changes between two adjacent data points, and hence results in some error.

	Avg. Error Per Data Point (BPM)	Avg. Error Between Data Avg. (BPM)
Set 1 Front	**2.97**	**0.46**
Set 1 Bottom	5.11	1.98
Set 2 Front	4.50	2.83
Set 2 Bottom	3.60	1.38
Sets 1 and 2	4.05	1.60

Table 1: Average error per data point and average error between data averages for front and bottom views of the first and second datasets

3.1.1 Incorporation of Computational Imaging

In this section, we present our results for different levels of camera control over ISO number and exposure time. We recorded our own video for different ISO numbers and applied our proposed image processing pipeline to estimate the average heart rate. Due to limitations in accessing resources, we were able to observe the average heart rate instead of heart rate variation in time. In Table 2, average heart rate estimates for different ISO numbers can be observed. Since increasing ISO number increases the brightness, we expected to see better estimates up to a point.



(a) The front view (the first set).

(b) The bottom view (the first set).

(c) The front view (the second set).

(d) The bottom view (the second set).

Figure 2: Extracted PPG signals for the provided datasets.

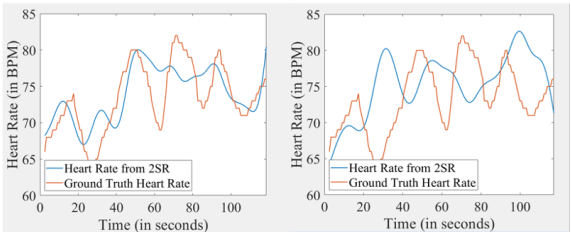


Figure 3: Computed heart rate vs. ground truth heart rate for the first dataset, front and bottom views respectively.

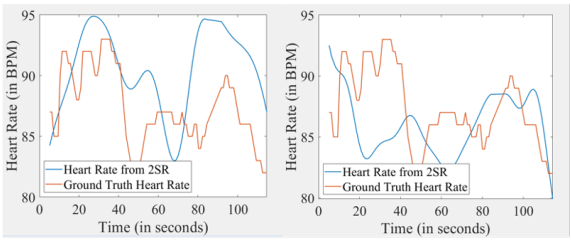


Figure 4: Computed heart rate vs. ground truth heart rate for the second dataset, front and bottom views respectively.

However, increasing ISO causes noise amplification as well. Therefore, we expected to observe deterioration in estimates when we increased the ISO number further. As it can be seen in Table 2, the results are consistent with our expectations and we obtained the best estimate when ISO number was 400. In the next experiment, we recorded three different videos by changing the exposure time. In particular, we recorded our videos under very low, normal and very high exposures. In order to control the exposure time of our recording, we used a third party app, however, we weren't able to see the exact exposure time in that app. Again, we applied our proposed pipeline to the videos and estimated the average heart rate. The results can be seen in Table 3. The average heart rate is underestimated in case of very low exposure time which is expected since the brightness was quite low. When we increased the exposure time, it improved our estimate and it became quite close to the ground truth. However, when we increased the exposure time further, our estimation deteriorated.

	Avg. Heart Rate Estimates
ISO 33	96.77
ISO 125	78.34
ISO 400	**70.22**
ISO 640	81.05
ISO 1000	60.01
ISO 1250	85.37
ISO 1600	61.03

Table 2: Average heart rate estimates for different ISO numbers (the ground truth is approx. 70 BPM)

	Low Exposure	Normal Exposure	High Exposure
Avg. HRE	48.25	**67.55**	29.18

Table 3: Average heart rate estimates for different exposure times (the ground truth is approx. 70 BPM)

4. Conclusion

In our experiments we were able to successfully estimate a subjects heart rate with respect to time. We built an algorithm pipeline around 2SR, and were able to calculate heart rate at any given time with an average error of around 4 BPM and calculate the average heart rate over an extended period of time with an average error of around 1.6 BPM (Table 1). With more control over the camera setup, this algorithm can be further improved by integrating computational imaging techniques. We demonstrated that there is an optimal ISO range in which our algorithm will perform its best (Table 2). Similarly, there is an optimal range for the exposure, so that the video is neither over nor under exposed and 2SR can properly extract the PPG signal (Table 3). The integration of more advanced computational imaging techniques, specifically spectral filtering and multiple exposure capturing, can be further analyzed and realized with access to the proper equipment.

References

[1] G. de Haan and V. Jeanne, "Robust pulse rate from chrominance-based rppg," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 10, pp. 2878–2886, 2013. [Online]. Available: https://doi.org/10.1109/TBME.2013.2266196

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

[3] W. Wang, S. Stuijk, and G. de Haan, "A novel algorithm for remote photoplethysmography: Spatial subspace rotation," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 9, pp. 1974–1984, 2016. [Online]. Available: https://doi.org/10.1109/TBME.2015.2508602

[4] M. Fabien, "A guide to face detection in python," Jul 2020. [Online]. Available: https://towardsdatascience.com/a-guide-to-face-detection-in-python-3eab0f6b9fc1?gi=4b054f467057

[5] D. E. King, "dlib-models," https://github.com/davisking/dlib-models, 2020.

[6] E. Buza, A. Akagic, and S. Omanovic, "Skin detection based on image color segmentation with histogram and k-means clustering," *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 1181–1186, 2017.

[7] C. S. Pilz, "Spatial-subspace-rotation," https://github.com/partofthestars/Spatial-Subspace-Rotation, 2017.

[8] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan, "Algorithmic principles of remote PPG," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 7, pp. 1479–1491, 2017. [Online]. Available: https://doi.org/10.1109/TBME.2016.2609282

[9] F. Scholkmann, J. Boss, and M. Wolf, "An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals," *Algorithms*, vol. 5, no. 4, pp. 588–603, 2012. [Online]. Available: https://doi.org/10.3390/a5040588

[10] M. Josemaria, "Ampd-algorithm," https://github.com/mathouse/AMPD-algorithm, 2014.

[11] A. V. Moço, S. Stuijk, and G. D. Haan, "New insights into the origin of remote ppg signals in visible light and infrared," *Scientific Reports*, vol. 8, no. 1, 2018.

[12] H. Voorhees and T. Poggio, "Computing texture boundaries from images," *Nature*, vol. 333, no. 6171, p. 364–367, 1988.