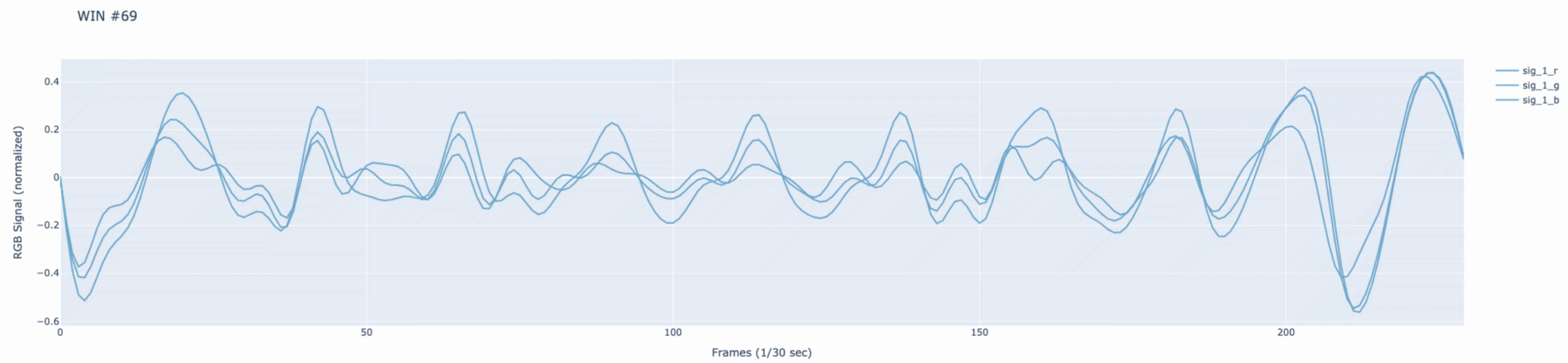
Real-time analysis of bio-signals from RGB streams on embedded systems

Romir Kulshrestha
r.kulshrestha@student.tudelft.nl

Kianoush Rassels Supervisor
Jorge Martinez Castaneda Responsible Professor

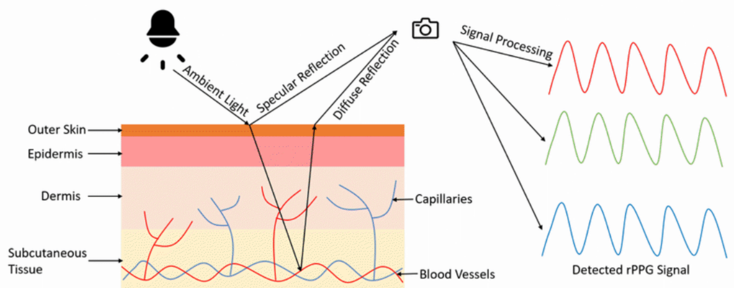


Above: Visualization of an RGB signal processed by rPPG to estimate a heart rate.

1. Background

The monitoring, detection, and analysis of these vital signs is crucial for timely medical intervention as well as the improvement of patient outcomes in a variety of settings, ranging from intensive care units (ICUs) to elderly care homes. However, conventional methods overwhelmingly rely on contact-based techniques that may cause discomfort or irritation in some patients or simply be too restrictive and impractical for long-term use.

Modern imaging systems are now inexpensive, ubiquitous, and designed to be integrated into more complex systems. This provides an avenue for the development of low-cost, embedded systems that utilize an RGB camera to remotely monitor vital signals. A number of analysis methods have been developed involving a variation of techniques and approaches to extracting a signal from a video feed. However, there is little to no research at the present moment on the real-time computational cost of running these analysis methods, especially in the case of low-cost embedded devices that do not have access to high-performance computational power.



Above: Principle of Remote Photplethysmography.

2. Research Question

What are the computational requirements of algorithms used for video-based biometric signal extraction?

3. Experiments

Algorithms benchmarked on realtime performance, memory, accuracy, and reliability while controlling for specific environmental factors. The ground truth is determined from a contact-based measurement from the UBFC rPPG Dataset.

3.1 Region of Interest

Determining and masking out the ROI can be a computationally heavy process. Three common methods—MediaPipe face detection, OpenCV Cascade Classifiers and skin segmentation—were evaluated on test hardware.

3.2 Runtime

This experiments compares the runtime performance of several prominent rPPG algorithms (PCA, PBV, POS and CHROM), examining their speed and resource demands as well as their accuracy and reliability. Understanding these performance characteristics is crucial for selecting the most suitable algorithm for real-time applications and resource-constrained environments.

5. Results

1) **Skin segmentation is unreliable**: Skin segmentation did not produce a usable ROI in 37.5% of the samples.

2) **MediaPipe offers the lowest benchmarked average time**: per frame of 9ms on a Raspberry Pi 5 (in comparison to 161ms and 254ms respectively taken by OpenCV and skin segmentation).

3) **ROI selection is a significant bottleneck**: Even when using MediaPipe, this step accounts for 96% of the computational load on average. In many cases, manual ROI selection may bepreferable to avoid this cost.

4) **CHROM and POS may be good candidates for further analysis**: POS allocates the most memory and runs for longer than all of the other studied algorithms. However, it also offers relatively higher accuracy and reliability. CHROM offers much lower memory usage and runtime for only slightly lower accuracy and reliability.

5) **Python shows promise**: as a candidate for real time embedded devices due to rich scientific ecosystem and decades of optimization.

6. Conclusion

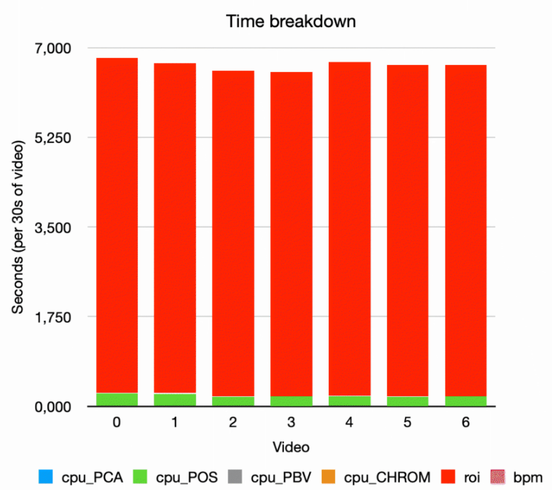
This research provides insights into the challenges of deploying real-time contact-free HR measurement systems. By studying the computational requirements of various estimation methods, it provides direction on where future research might be concentrated.

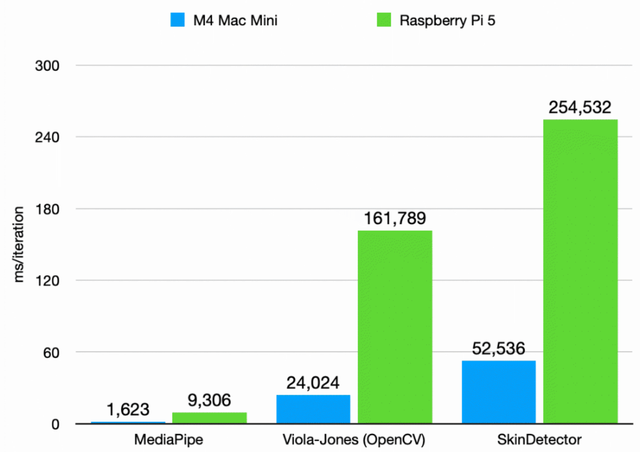
Despite advances in contact-free estimation methods, achieving real-time performance on real-time embedded systems remains a challenge. ROI selection poses a significant bottleneck—therefore, optimizing ROI selection is a good first step to implementing contact-free HR measurement on embedded systems.

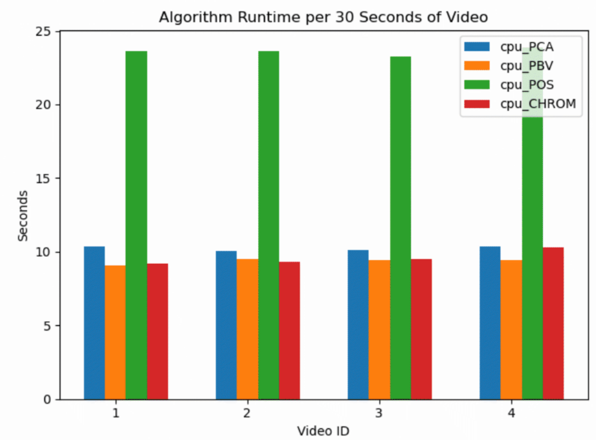
CHROM appears to offer a good balance of low computation cost and accuracy. POS could be a viable alternative as well. Future research may explore optimizing these algorithms to bring down computational cost while improving accuracy and reliability.

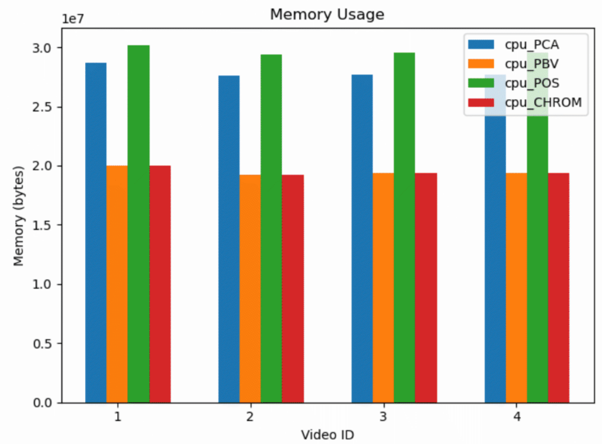
Future studies should expand the scope of algorithms and platforms considered, as this short study is limited to mainly Python-based implementations and a limited set of techniques.

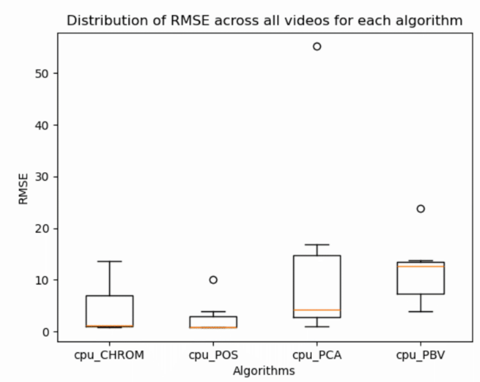
Additional work is required to test implementations in live scenarios, subject to unpredictable and dynamic real-world environments.











From left to right: Time breakdown of processing steps per 30s of raw video; Runtime comparison of ROI selection algorithms; Runtime and peak memory usage of rPPG methods; RMSE scores of rPPG methods