

VideoMotion Project

Overview

The VideoMotion project is a platform to see if human heart rate can be determined by analyzing micro changes in face motion and color over time. Research suggests that as the heart pumps blood into the head, there are small but perceptible changes in the head vertical position and also changes in the green color as blood hemoglobin and oxyhemoglobin absorb light.

Operation

To use the application, download and configure the application from GitHub and run as follows:

```
python main.py                (To estimate heart rate from a camera attached to the computer)
python main.py video_file     (To estimate heart rate from a video file)
```

The configuration file, config.txt contains various options. See the README.md file in the github repository for details

Methods for Facial detection and tracking

The open source project “OpenCv” is used to detect a face and track the face changes over a period of time. Currently this uses only frontal face classifiers however side profile classifiers are available.

An option in config.txt allows the operator to manually select an rectangular region of interest., (ROI) instead of face detection. (This is useful for analyzing video files)

Methods used to measure facial changes

Two methods are used to measure facial changes related to heart rate.

1. Changes in the vertical head position that result from the heart pumping blood into the head
2. Changes in the green color as blood flows to and from the face

Methods used to extract heart rate signal

The time series from the above are band pass filtered to isolate signals in the heart rate range. The bandpass filter is 42 to 150 bpm. (This is configurable via config.txt)

Methods used to estimate heart rate from the signal

Method 1 uses a peak detector on the time series to determine the heart beat cycle. Method 2 uses an FFT to isolate the dominant frequency

A composite heart rate is determined by summing the FFTs of the color and motion signals then selecting the dominant frequency

Calibration

The results measured are currently compared to results from a fit bit. Fit bits are not considered very reliable heart rate trackers so calibration definitely needs attention, especially in cases where the fit bit data is somewhat 'close' to the estimated data.

Error rates

Error rates have not been quantified, however empirically it appears they are quite high: In the order of +20 BPM

Video Properties

Most of the estimations used 640 X 480, 30 FPS video. The literature indicates that 30 FPS is more than sufficient for the approximately 1 Hz heart rate. Higher frame resolutions would probably improve results, although at the expense of processing time.

Empirically it appears that the larger the face area, the more consistent results

Performance

The first phase of this project is to optimize the results without regard to performance. However as a base indication, a MacBook can process about 20 FPS while a Raspberry Pi 3+ can process 3-4 FPS. Note that the core computations occur in OpenCV which is implemented using C++. The actual python processing overhead is TBD

'Motion' vs 'Color'

Empirically, color processing appears to produce more consistent 'dominant' frequencies than motion. Also when a strongly dominant frequency is detected, it appears to be that of the heart rate.

The heart rate BPM reported on the video is the dominant frequency calculated by summing the motion and color FFTs.

Noise sources

The measurement of heart rates is affected by head motion 'noise' in the motion dimension and ambient light 'noise' in the color dimension

Heart rate inter-beat interval

This project does not support any analysis of heart rate inter-beat intervals

Things to do to improve accuracy?

- Remove effects of 'large' motion/color changes. (A large step function may introduce filter ringing and as such add lower harmonics in the FFT)
- Reduce Heart rate range. Currently 42 BPM to 150 BPM

- Combining the color and motion results currently uses arithmetic summation on the 2 FFTs and selecting the maximum. (This may be different than correlating the two time series, summing the result and calculating both the peak values and the FFT)
- Quantify 'noise' sources and remove 'noise' from heart rate signal

Example:

The chart below shows heart rate measured at 10 second intervals over 6 minutes. A Macbook pro web-cam using 640 *480, 30 FPS was used. The test subject was between 9" and 24" from the camera. The Fitbit used as a reference show heart rate range of 54 to 61 BPM. The composite FFT and color FFT HR estimates show less variance than the motion and pk-pk methods

