Flowchart of steps involved in the algorithm

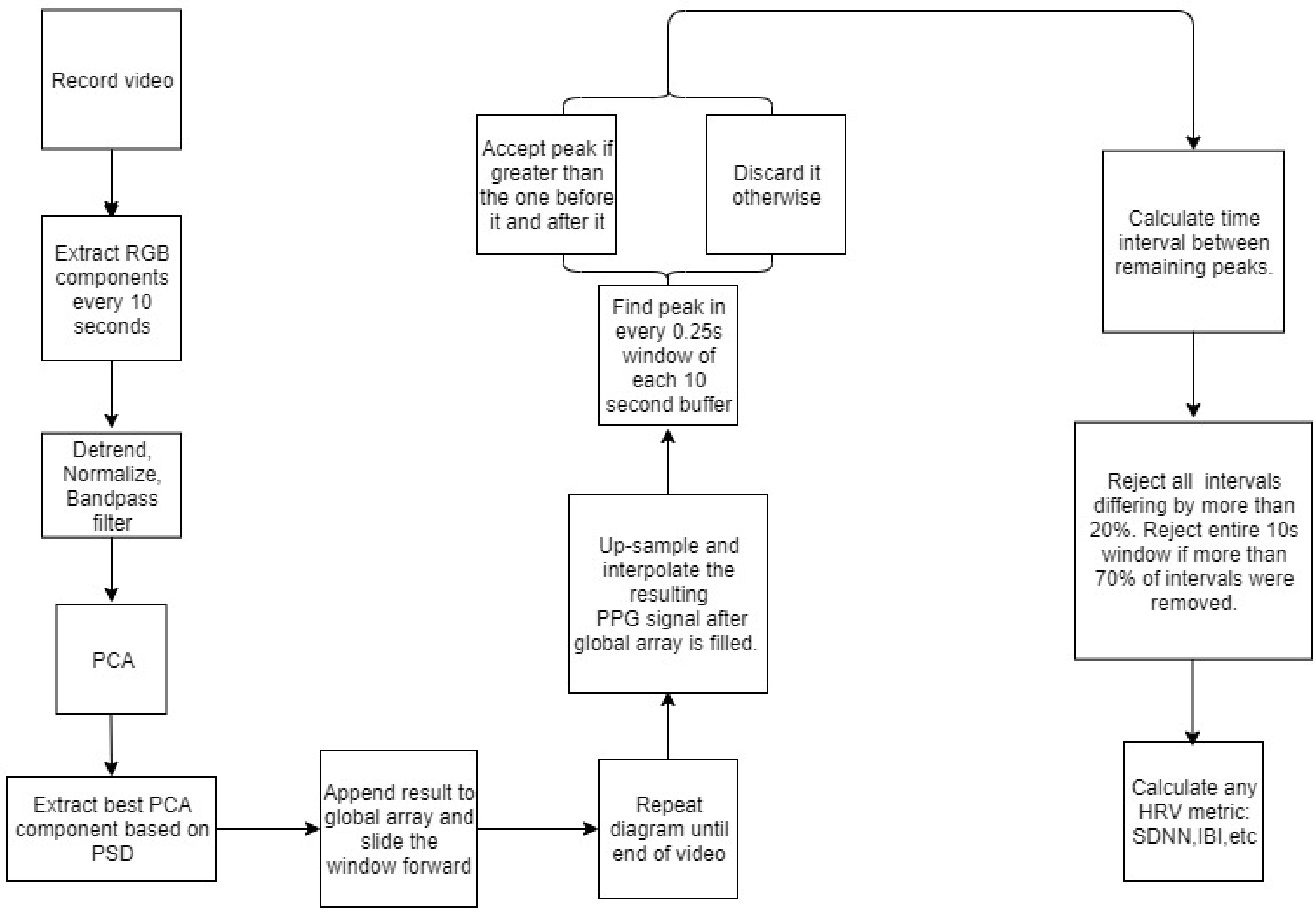


Figure 1: Position for video recording

<u>Step 1:</u>

A video is recorded with the subject positioned as seen in figure 1. In this case the camera recorded at 30 fps, and the video length was 3 minutes→5400 frames. The video was illuminated with artificial lighting

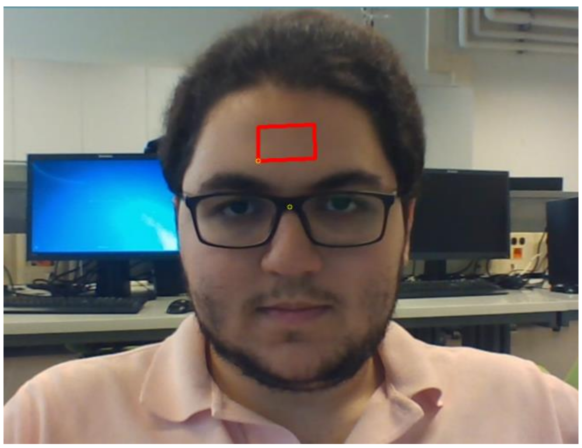


Figure 2: ROI detection

<u>Step 2.A:</u>

Using OpenCV and a pre-trained model file, facial landmarks are detected (nose, eyes, mouth, eyebrows) in every frame. By locating the point between the eyebrows and offsetting upwards, we trace the red rectangle seen in figure 2, the inside of which will be our region of interest (ROI). Everything outside this rectangle is discarded

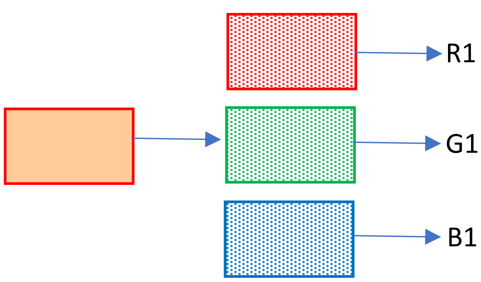


Figure 3: separating the ROI into color components

<u>Step 2.B:</u>

The ROI is separated into 3 components: Red, Green, and Blue

Each component is spatially averaged, to finally get a single value for each of the 3 colors per frame.

e.g. R1 is the red value for frame 1.

We do this for every frame until we have collected 10 seconds worth (i.e. 300 frames)

Figure 4

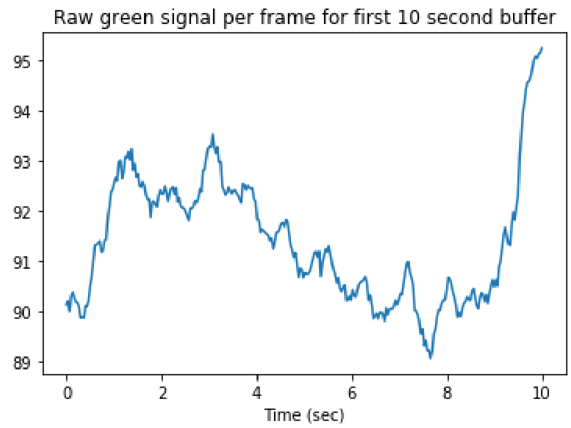
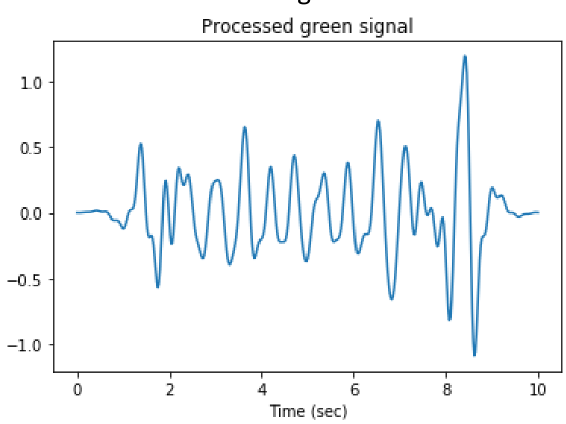


Figure 4 illustrates the first 10-second buffer for the green signal of the given video, without any processing performed. Note that we are working with 3 colors and not just Green, however this figure is just to illustrate how things look like at this point.

Figure 5



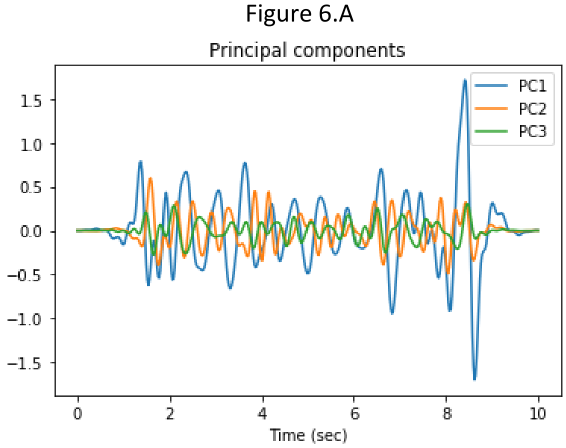
<u>Step 3:</u>

We process each of the 3 buffers as follows:

- Detrend using smooth priors approach with λ = 2000
- Normalize (subtract mean and divide by std)
- Convolve with 128-point hamming window bandpass filter having as bound 0.7 Hz and 3 Hz.

Figure 5 shows the green signal previously illustrated after it is processed using these 3 steps.

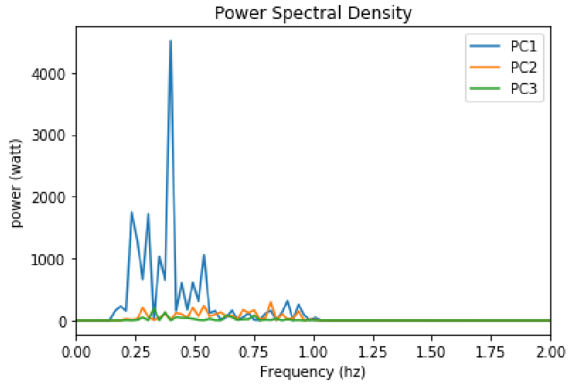
Figure 6.A



<u>Step 4:</u>

Next, we perform PCA, with the main purpose of minimizing correlation between the 3 color signals without compromising the variance of the data.
The result is 3 principal components, which are a linear combination of the 3 color signals (RGB), with the first component having maximum variance.
Figures 6.A shows the 3 principal components obtained from the first 10-second buffer.

Figure 6.B



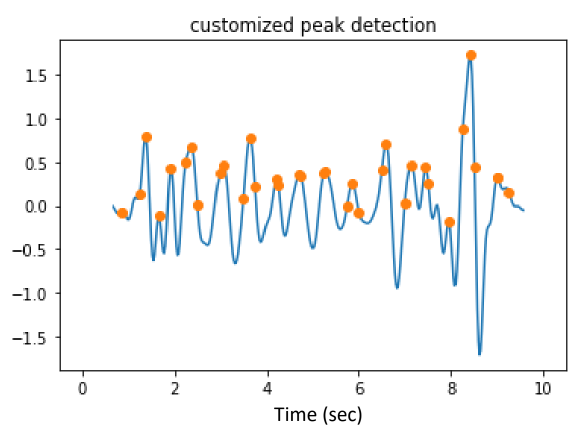
<u>Step 5:</u>

We plot the Power Spectral Density of the 3 principal components and chose the one with the maximum peak. In this case, as shown in figure 6.B, it is definitely the first component.

This component will be our PPG signal for the 10-second intervals we have been working with.

Since we are working with an offline video in this case, we repeated this whole process till the end of the video, storing at each iteration the extracted PPG signal in a global array, and moving the 10 second buffer by increments of 3 seconds. That is: we truncate the first 3 seconds from the R,G,B buffers, and add to them new 3 seconds of footage. This counts as steps 6 and 7 in the flowchart shown above. In a real-time setting however, this 10-second PPG signal is enough to extract some information about cardiovascular activity. As such, the rest of the explanation will utilize this 10-second buffer alone instead of the whole video.

Figure 7

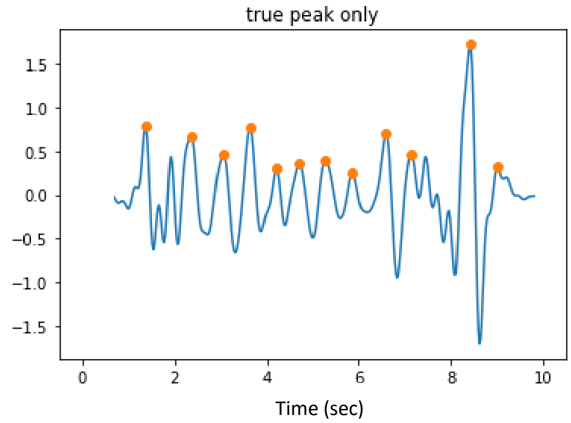


<u>Step 8:</u>

We interpolate the PPG signal to a frequency of 240 Hz.

Then we apply a customized peak detection algorithm, wherein we locate the peaks using a 0.25 second sliding window. The obtained peaks for our example are given in figure 7.

Figure 8

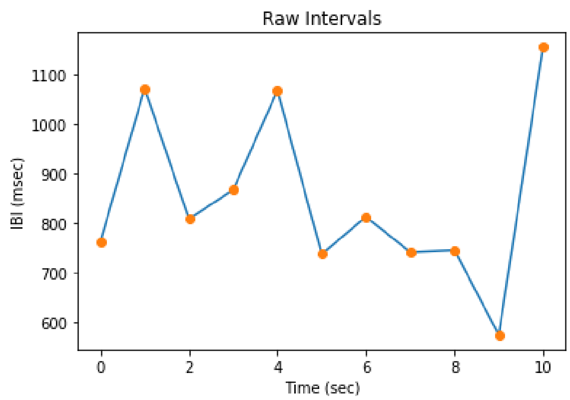


Step 9:

The next step is to discard all the false peaks, by doing the following:

If a peak is greater in magnitude than the one before it and the one after it, it is a true peak. If not, it is a false one, and is discarded. Figure 8 shows the true peaks remaining after performing this step.

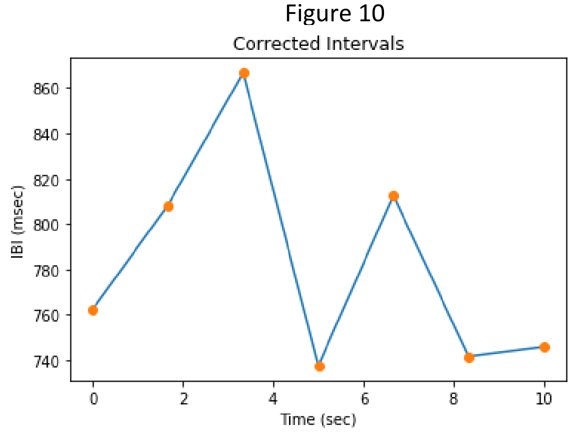
Figure 9



Step 10:

From the true peaks, we can calculate the inter-beat intervals.
We know the frame at which each peak occurs. For each 2 consecutive peaks, we subtract their frame location and divide the difference by the sampling rate, 240 Hz in this case, to get the IBI in seconds.
Doing this over the whole buffer gives us 11 intervals in total, which can be seen in Figure 9.

Figure 10



Step 11.A:

Nevertheless, these intervals are further corrected as follows:

Discard any interval that differs by more than 20% from the median of all intervals obtained from this buffer. The median in our example is equal to 808.3 msec.

This step leaves us with 7 intervals from the original 11, as shown in figure 10.

Step 11.B:

As a final step, we check if the number of remaining intervals after this correction is more than 30% of the uncorrected intervals.

In this case 30% of 11 is 3.3, whereas we have 7 intervals remaining after correction, so this particular buffer passes this check.

If not, then we would've discarded the whole buffer as too noisy for HRV analysis.

Step 12:

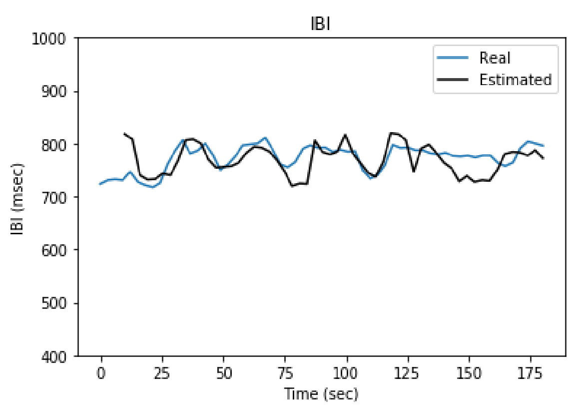
Now, we can easily calculate an average IBI reading for this 10-second buffer, which is equal to 782.14 msec.

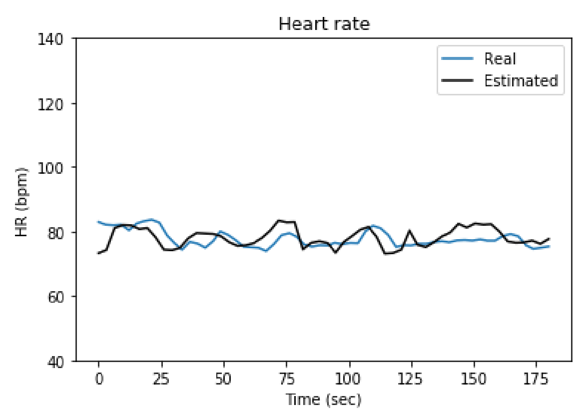
We also calculate the heart rate as $60/IBI_{average.}$

In this case, HR = 76.7 bpm

We have shown above how we have extracted information about cardiovascular activity from a pre-recorded video, noting that the example was applied to 10 seconds worth of footage only. Nevertheless, this procedure can be reproduced for the whole video by advancing the 10 second buffer every 3 seconds.

As a result, we can obtain a heart rate and IBI reading every 3 seconds. For the video used in our example, the following results were obtained:





Recap of summary above: