HeartPy: A novel heart rate algorithm for the analysis of noisy signals

van Gent, Paul; Farah, Haneen; van Nes, Nicole; van Arem, Bart

DOI
10.1016/j.trf.2019.09.015

Publication date
2019

Document Version
Final published version

Published in
Transportation Research Part F: Traffic Psychology and Behaviour

Citation (APA)
van Gent, P., Farah, H., van Nes, N., & van Arem, B. (2019). HeartPy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation Research Part F: Traffic Psychology and Behaviour*, *66*, 368-378. https://doi.org/10.1016/j.trf.2019.09.015

Transportation Research Part F 66 (2019) 368–378

Contents lists available at ScienceDirect

Transportation Research Part F

journal homepage: www.elsevier.com/locate/trf

HeartPy: A novel heart rate algorithm for the analysis of noisy signals

Paul van Gent [a,*], Haneen Farah [a], Nicole van Nes [b], Bart van Arem [a]

[a] Department of Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology. Stevinweg 1, 2628CN Delft, the Netherlands
[b] Stichting Wetenschappelijk Onderzoek Verkeersveiligheid (SWOV), Bezuidenhoutseweg 62, 2594AW Den Haag, the Netherlands

ARTICLE INFO

Article history:
Received 30 September 2018
Received in revised form 5 August 2019
Accepted 24 September 2019

Keywords:
Human factors
Heart rate analysis
Physiological signals
Signal analysis
Open source

ABSTRACT

Heart rate data are often collected in human factors studies, including those into vehicle automation. Advances in open hardware platforms and off-the-shelf photoplethysmogram (PPG) sensors allow the non-intrusive collection of heart rate data at very low cost. However, the signal is not trivial to analyse, since the morphology of PPG waveforms differs from electrocardiogram (ECG) waveforms and shows different noise patterns. Few validated open source available algorithms exist that handle PPG data well, as most of these algorithms are specifically designed for ECG data.

In this paper we present the validation of a novel algorithm named HeartPy, useful for the analysis of heart rate data collected in noisy settings, such as when driving a car or when in a simulator. We benchmark the performance on two types of datasets and show that the developed algorithm performs well. Further research steps are discussed.

1. Introduction

Vehicle automation is rapidly gaining popularity in the agendas of the automotive sector and governments. Automation promises to increase traffic flow efficiency (Hoogendoorn, van Arem, & Hoogendoorn, 2014) and free up the time of the driver for other activities. However, (semi-)autonomous vehicles below SAE level 5 will still need to interact with the driver, for example for a transition of control, or in emergency situations when the automation fails. This means that the vehicle needs to be aware of the driver's state (e.g. distraction, fatigue), because a transition of control can be dangerous when the driver is not able to take over control of the vehicle (Merat, Jamson, Lai, Daly, & Carsten, 2014), because of for example high workload or distraction. Additionally, In-Vehicle Information Systems (IVIS) that interact with the driver to provide information or advices can also benefit from knowledge about the driver's state to choose the most appropriate interface (Birrel, Young, Stanton, & Jennings, 2017; Park & Kim, 2015), content and timing of the information (van Gent, Farah, van Nes, & van Arem, submitted for publication).

Algorithms are being developed that can estimate the driver's state, whether this is driver workload (Solovey, Zec, Garcia Perez, Reimer, & Mehler, 2014; van Gent, Melman, Farah, van Nes, & van Arem, 2018), driver distraction (Liang, Reyes, & Lee, 2007) or a driver's interruptibility (Kim, Chun, & Dey, 2015). Heart rate is frequently included as an input for predicting a

* Corresponding author.
E-mail addresses: P.vanGent@tudelft.nl (P. van Gent), H.Farah@tudelft.nl (H. Farah), Nicole.van.nes@swov.nl (N. van Nes), B.vanArem@tudelft.nl (B. van Arem).

https://doi.org/10.1016/j.trf.2019.09.015
1369-8478/

driver's state since it contains information about changes in (driver) workload (Mehler, Reimer, & Coughlin, 2012; Mehler, Reimer, Coughlin, & Dusek, 2010), stress (Healey & Picard, 2005), and general driver state such as drowsiness (Danisman, Bilasco, Djeraba, & Ihaddadene, 2010). In addition to the benefits for autonomous systems, many human factors studies focusing on the interaction between the driver and (semi-)autonomous vehicles also include heart rate measurements (Jamson, Merat, Carsten, & Lai, 2011; Reimer, Mehler, & Coughlin, 2016; Reimer, Mehler, Coughlin, Roy, & Dusek, 2011; Stapel, Mullakkal-Babu, & Happee, 2017).

However, capturing heart rate in the often noisy conditions of either a driving simulator or in an in-vehicle setting, and subsequently analysing the complex signals either real-time or offline, can be difficult or costly (Brookhuis & de Waard, 2010). Low-cost commercial devices are available, but these are generally designed for sporting contexts and not specifically for scientific research. Furthermore, the proprietary nature of the firmware and software used in these devices creates problems with data reliability, reproducibility of results, and integration into in-vehicle hardware for the purpose of real-time driver monitoring. This reduces the usefulness of these devices for research into (partially) self-driving vehicles and makes it nearly impossible to integrate them into actual in-vehicle systems.

One potential solution lies in the recent advances in wearable technology and open hardware platforms, such as Arduino[1] and Raspberry Pi.[2] They are open source in both hardware and software design, meaning that integration into existing systems is feasible. There is, however, a lack of open source available heart rate analysis algorithms that are validated, easy to use and able to handle noisy data from low-cost PPG sensors. Implementations of heart rate analysis algorithms described in research papers are often not available, poorly documented, or require substantial technical expertise to implement properly.

In a previous study we collected heart rate data with low-cost sensors to develop an affordable driver workload estimation approach (van Gent et al., 2018; van Gent, Farah, van Nes, & van Arem, 2017). Available open source algorithms did not function well on this type of -often noisy- data and couldn't easily be integrated into a real-time system. To overcome this issue, our aim is to develop a novel algorithm that (i) functions better on this type of noisy data, and (ii) provides an easy-to-use analysis method for the collected heart rate data both, offline and real-time. We've named the developed algorithm HeartPy. For a technical overview of HeartPy, its development and its availability, please see (van Gent, Farah, van Nes, & van Arem, submitted for publication). The main aim of this paper is to describe the validation of HeartPy using two datasets: a noisy dataset collected in a driving simulator (van Gent et al., 2017), and an openly available medical dataset (Jager et al., 2003).

In the rest of this paper, we first describe basic properties of the heart rate signal as they relate to data collection and analysis. This is followed by a brief overview of the algorithm's functioning, discussion of our methods, results and concluding remarks.

1.1. Measuring heart rate in naturalistic or simulated settings

There are two major approaches to measuring heart rate, which mainly differ in the physiological properties they measure.

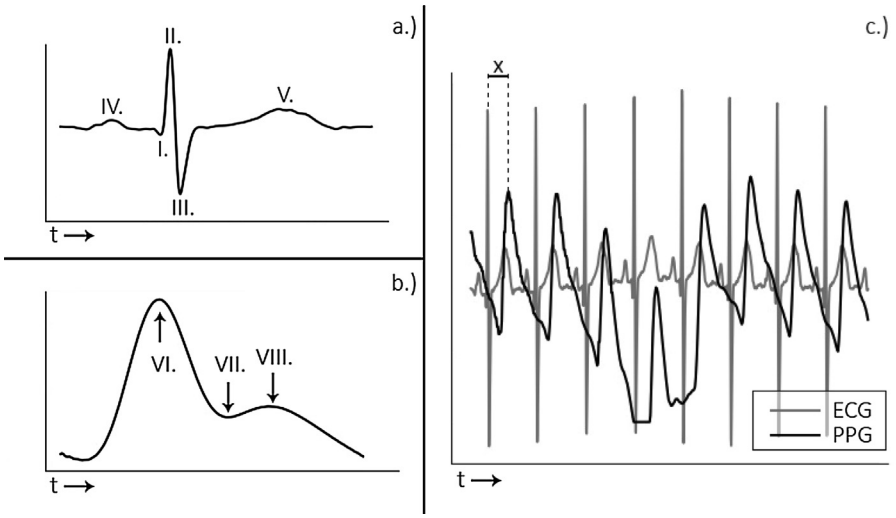

Fig. 1. The differences in morphology of the ECG wave (a) and PPG wave (b), and the time lag 'x' between both waves (c). The ECG (a) wave consists of most notably the Q-R-S complex (I-III). The P (IV) and T (V) waves are also marked in the plot. The PPG (b) wave consists of the systolic peak (VI), the diastolic peak (VIII) and the dichrotic notch (VII).

[1] See http://www.arduino.cc.
[2] See http://www.raspberrypi.org.

Electrocardiogram recordings (ECG) are collected by placing electrodes on the chest near the heart. These electrodes measure the electrical activation of the heart during each cardiac cycle. The defining feature in the ECG signal is the QRS complex (Fig. 1a I-III). Advantages of the ECG signal are that it directly measures the heart's electrical activation and that it presents a strong QRS complex presence in the resulting signal (Fig. 1a). A common source of noise in ECG signals are motion artefacts resulting from sensor displacement due to participant movement. These tend to fall in the same frequency range as the QRS-complexes, which can make it difficult to filter them without deforming the QRS complex (Kirst, Glauner, & Ottenbacher, 2011). In traffic related studies, ECG recordings have been used in for example (Brouwer & Dijksterhuis, 2015; Fallahi, Motamedzade, Heidarimoghadam, Soltanian, & Miyake, 2016; Farah et al., 2012; Miyaji, Danno, Kawanaka, & Oguri, 2008). The main disadvantage is that ECG is not easily measured unintrusively.

Photoplethysmogram (PPG) recordings offer a less invasive method of assessing the cardiac cycle. These devices employ an optical sensor to measure the changes in coloration of the skin as blood perfuses through the arteries and capillaries with each heartbeat. PPG is typically measured at the fingertip or through wrist bracelets. The PPG signal consists of a systolic peak (Fig. 1b-VI), a dicrotic notch (1b-VII), and a secondary peak called a diastolic peak (1b-VIII). The secondary peak may be absent in some recordings or of very low amplitude. Advantages of the PPG method are that it is low cost, easy to set up, and non-invasive (Elgendi, 2012; Millasseau et al., 2000). Ways of obtaining the PPG signal contactless through cameras have been demonstrated, further reducing intrusiveness (Sun, Hu, Azorin-Peris, Kalawsky, & Greenwald, 2012). However, PPG tends to display more amplitude variation over short time-intervals (Fig. 1c), more variation in waveform morphology, as well as contain more noise from various sources when compared to ECG measurements. This makes analysis more difficult. In the traffic domain, PPG sensors have been used by for example (Jarvis, Putze, Heger, & Schultz, 2011; van Gent et al., 2018; Zhai & Barreto, 2006).

1.2. Analysing heart rate data

The heart signal is often split into heart rate (HR) and heart rate variability (HRV) measures. Heart rate is a simple measure of the heart period, expressed in the beats per minute and the inter-beat interval. Heart rate variability measures describe how the heart rate signal varies over time, and can be divided into time-domain measures and frequency-domain measures (Mehler, Reimer, & Wang, 2011; Montano et al., 2009).

When extracting heart beats from a signal, a marker is chosen that can reliably be detected at the same position on all heartbeat complexes in the signal. In the ECG the R-peak is often taken (Fig. 1a-II), in the PPG signal the maximum of the Systolic wave is usually marked (Fig. 1b-I).

Common measures expressing the HR found in the literature are the beats per minute (BPM) and the mean inter-beat interval (IBI). HRV is expressed by the median absolute deviation of intervals between heart beats (MAD), the standard deviation of intervals between heart beats (SDNN), the root mean square of successive differences between neighbouring heart beat intervals (RMSSD), the standard deviation of successive differences between neighbouring heart beat intervals (SDSD), and the proportion of differences between successive heart beats greater than 50 ms and 20 ms (pNN50, pNN20, resp.). HRV can also be expressed in the frequency domain, where two frequency bands are usually included: low frequency (LF, 0.04–0.15 Hz), which is related to short-term blood pressure variation (Bernardi et al., 1994), and high frequency (HF, 0.16–0.5Hz), which is a reflection of breathing rate (Montano et al., 2009).

Despite the different underlying physiological constructs that are measured, a high correlation (median 0.97) between peak-peak intervals extracted from ECG and PPG signals has been reported (Selvaraj, Jaryal, Santhosh, Deepak, & Anand, 2008). This makes the PPG a valid alternative for applications that require non-intrusive heart rate measurements, given that validated analysis algorithms exist.

1.3. Development and availability HeartPy

We developed HeartPy to help analyse noisy heart rate data collected in driving settings (both simulated and on-road). The algorithm runs on desktop computers (Python) as well as wearables (embedded C) such as Arduino and Teensy boards, both offline and in real-time. The latter allows for real-time heart rate analysis in in-car settings as well as other mobile situations, such as with cyclists or pedestrians. The algorithm is available as the Python package 'HeartPy', hosted on GitHub (van Gent, 2017) and is installable through Python's 'pip' package manager as well. Documentation is available through the GitHub page. The wearable embedded C version is available on GitHub as well (van Gent, 2018), together with documentation linked there.

HeartPy was designed to be resistant to typical noise patterns (e.g. motion artefacts, momentary signal loss) of participants engaged in other tasks (driving simulator, on-road car experiment, bike experiment), to be capable of handling signals from low-cost off-the-shelf sensors, and to be user friendly.

1.4. Overview of the HeartPy algorithm

HeartPy comes with various pre-processing options to clean up signals, including FIR filtering and outlier detection. This section briefly outlines the peak detection methods. Please refer to van Gent et al. (submitted for publication) for more information on the software, its availability and its functioning.

Peak detection uses an adaptive threshold (Fig. 2III) to accommodate for morphology and amplitude variation in the PPG waveform, followed by outlier detection and rejection. To identify heartbeats, a moving average is calculated using a window of 0.75 s on both sides of each data point. Regions of interest (ROI) are computed between two points of intersection where the signal amplitude is larger than the moving average (Fig. 2I-II), which is a standard way of detecting peaks. Two methods of obtaining a peak's location are included. In the first approach, the peak position is simply taken to be the highest point in the marked ROI. Although this is a computationally low-cost operation, its accuracy depends on the sampling rate used, with a higher sampling rate resulting in more accurate results. In the second method a univariate spline is used to upsample and interpolate the ROI, which is then solved for its maximum. This requires more computation but is also more accurate, especially with lower sampling rates. Both methods are available in HeartPy, by default the fast method is used.

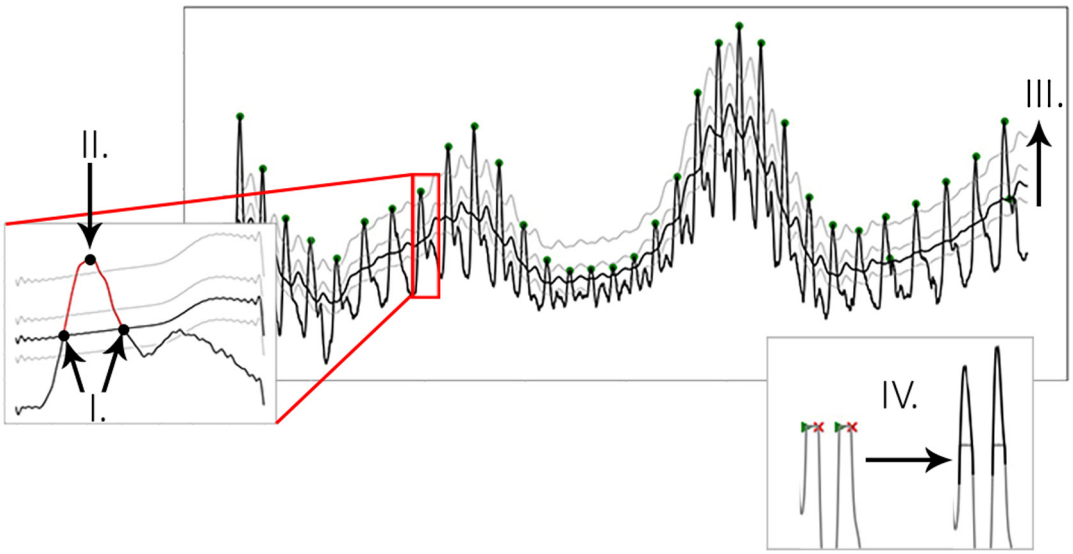

Fig. 2. Process of peak extraction: a moving average is used as an intersection threshold (I). Candidate peaks are marked at the maximum between intersections (II), with optional spline interpolation available to improve position accuracy. The moving average is raised stepwise (III). IV. shows the detection of the onset and end of clipping, and the result after interpolating the clipping segment.

Signal clipping is a special case that hinders the accurate placement of a peak's position. Clipping can occur for various reasons, for example when digitising an analog signal. HeartPy detects the onset and end of clipping segments, and will attempt to reconstruct the waveform by spline interpolation, as shown in Fig. 3IV.

During the peak detection phase, the amplitude of the calculated threshold is adjusted stepwise (Fig. 2III). The best fit is determined by minimising the standard deviation of peak-peak intervals (SDNN, see also 2.2). The instantaneous heart rate (BPM) is computed and evaluated in tandem with the SDNN. This represents a fast method of approximating the optimal threshold amplitude by using the (relative) regularity of the heart rate signal. As shown in Fig. 3, missing one peak (III.) already leads to a substantial increase in SDNN compared to the optimal fit (II.). Marking incorrect peak positions also leads to an increase in SDNN (I.). The lowest SDNN value that is not zero, in combination with a reasonable BPM value, is selected. The BPM must lie within a predetermined range (default: 40 <= BPM <= 180, range settable by user).

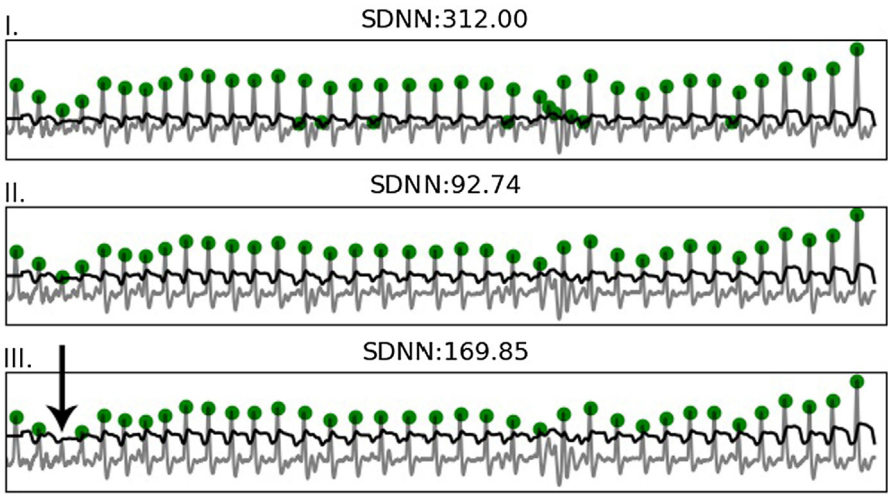

Fig. 3. Effects of missing or additional peaks on the SDNN metric: the last image (III.) shows that missing a single beat will already lead to a large increase in SDNN compared to the optimal fitting. BPM is also considered when fitting.

Due to the variable PPG waveform morphology, it is possible that after the initial peak fitting phase incorrectly marked peaks remain. Motion artefacts may be another cause of detection error. A correction is performed by thresholding the sequence of peak-peak intervals. Peaks are considered low confidence if the interval created between two adjacent peaks deviates by more than 30% of the mean peak-peak interval of the analysed segment (Fig. 4). The threshold is adaptive based on the current segment with a minimum value of 300 ms. We've found this to be a good approximation for incorrect detections. If any peaks are considered incorrect detections, the array of peak-peak intervals is recomputed to only contain intervals between two high confidence peak positions.

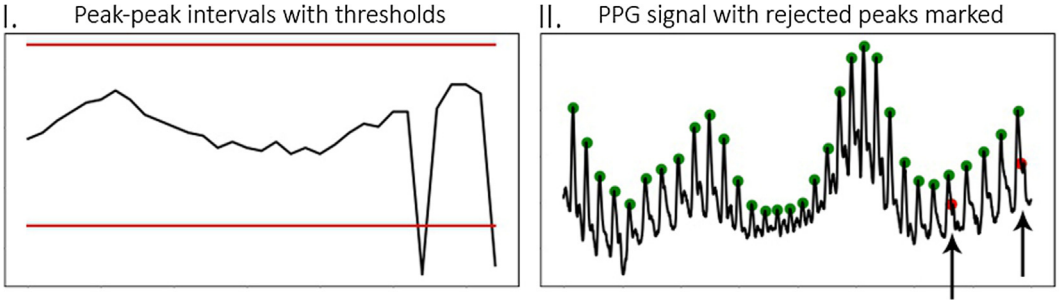

Fig. 4. The plotted peak-peak intervals with thresholds (I.), and the resulting rejected peaks (II.)

2. Methods

The algorithm was validated using two datasets from two different experiments and research domains. The first dataset used was collected with a low-cost PPG sensor in a driving simulator experiment (van Gent et al., 2018). This dataset contains approximately 20.7 h of PPG recordings. The second dataset is the openly available Long-Term ST Database (Jager et al., 2003), containing 86 long ECG recordings of 80 participants, with each recording being between 21 and 24 h.

The PPG dataset was used in its entirety and split into one-minute segments. Because the ECG dataset was so large, 1.000 one-minute segments were randomly selected from the database. The peak positions in all the segments from both datasets were annotated manually and checked a second time to serve as a ground truth. These annotations are also available on the GitHub page (van Gent, 2017). For both data sets a one-minute length for the segments was used to balance both the number of peaks in each segment with the time needed to manually annotate all segments. The algorithm performance was compared to the annotated data on four variables: detected peak position, mean of the peak-peak intervals calculated over the analysed segment, beats per minute computed by the algorithm (a HR measure), and a common heart rate variability (HRV) measure: the standard deviation of successive differences (SDSD). To quantify the accuracy of the algorithm's predictions, we used the Root Mean Squared Error (RMSE), defined as:

$$RMSE = \sqrt{\frac{\sum (y - \hat{y})^2}{n}} \quad (1)$$

where y is the ground truth value, $\hat{y}$ is the value predicted by the algorithm, and n the number of comparisons. For the accuracy of the absolute peak-positions in time as compared to the annotated ground truth, we used the mean of the absolute deviations.

2.1. Error types

In addition to the performance comparison, the results of the one-minute segments were plotted and three types of errors annotated (shown in Fig. 2): 'Incorrectly rejected' – meaning that a correct peak has been marked as low confidence (Fig. 2a). 'Missed' – indicating that a peak is present but not marked (Fig. 2b). 'Incorrectly accepted' – indicating a peak is marked where no peak is considered present by the human annotator (Fig. 2c). Fig. 2d, e and f show other examples that were classified as 'incorrectly accepted': cases where a peak was marked at a non-maximum position, or where a diastolic peak was marked instead of a systolic peak. The algorithm has been designed to minimise the 'incorrectly accepted' error type for reasons discussed in the next section.

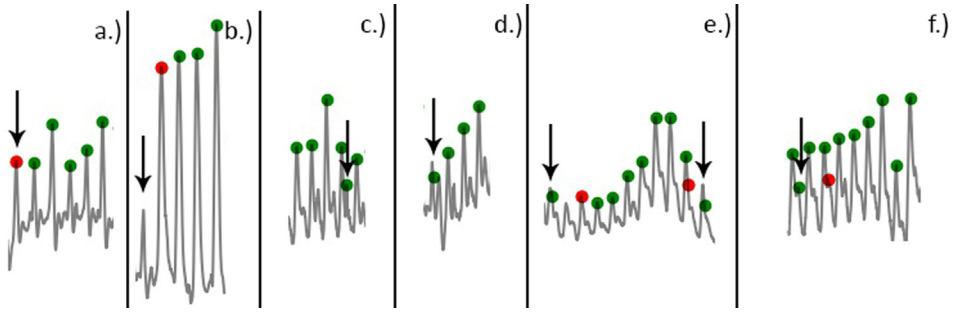

Fig. 5. Figure displaying the possible errors. These are: (a) 'incorrectly rejected', (b) 'missed', (c) 'incorrectly accepted'. Peaks marked on a correct QRS complex but not on its peak maximum, are also counted as 'incorrectly accepted'. This type of error is shown in (d). Other possible mistakes counted as 'incorrectly accepted' are marking a peak at a non-maximum position (e), or incorrectly marking a diastolic peak (f).

2.2. Minimising the correct error type

The algorithm was designed to minimise the 'incorrectly accepted' peak error types because this error type has the strongest effect on calculated output measures. This section illustrates why the choice was made.

Heart rate variability (HRV) measures are not robust against outliers. Marking a peak on an anomalous position affects these measures since they express the variation in the intervals between peak positions. Marking a peak at an incorrect time position creates a deviation in the length of the surrounding intervals which will strongly influence the variance in the sample. Note that the heart rate (HR) measures such as IBI and BPM are quite resistant to outliers because they use the mean of all peak-to-peak intervals in a given signal segment.

To further explain and show these effects, a bootstrapped simulation was performed. We took a manually annotated one-minute segment of PPG heart rate data and artificially introduced two types of errors:

(1) "Incorrectly rejected" peaks were simulated by dropping a random n% of peaks from the signal. Measures were then calculated on intervals between peaks where no missing value occurred in between, mimicking the algorithm's behaviour of only computing peak-peak intervals between two accepted peaks.
(2) "Incorrectly accepted" peaks were simulated by introducing a position error into a random n% of all peaks. The error disturbed the peak position randomly by 0.1–10%, meaning a random positional disturbance of between 1 ms and 100 ms. For each selected peak the disturbance magnitude was randomised.

Simulations were run with values of 5%, 10% and 20% for 'n'. Each simulation run was bootstrapped for 10.000 iterations to reduce the effects of the random selection process.

Results show that the effect of incorrect beat detections (displacement scenario) is significantly stronger than the effect of missing values, especially the effect on HRV measures. The effect on BPM is negligible in both the missing and displacement simulations, showing the HR measures' resilience to outliers. Effects on HRV measures are substantial. The data are displayed in Table 1 below, and the analysis notebook is available on the Python GitHub (van Gent, 2017).

[3] See: https://github.com/jramshur/ECG_Viewer.

Table 1
The absolute errors introduced in the simulations. The errors introduced in the HRV measures by displaced peaks are particularly evident (marked grey).

Errors introduced in simulated scenarios (n=10.000)						
	Missing n% of peaks			Displacing n% of peaks		
	n=5%	n=10%	n=20%	n=5%	n=10%	n=20%
BPM	0.001	0.007	0.062	0.000	0.000	0.000
IBI	0.012	0.067	0.541	0.002	0.004	0.003
SDNN	0.004	0.042	0.267	2.635	5.172	10.006
SDSD	0.614	1.210	2.354	9.543	16.365	26.200
RMSSD	0.773	1.565	3.161	10.774	19.658	34.446
pNN20	0.006	0.012	0.025	0.043	0.082	0.149
pNN50	0.005	0.010	0.022	0.064	0.125	0.232

The data in the table show that the errors induced are especially large in the case of the variability measures. As discussed, this has to do with what the measures are designed to express: the variability measures express the variation in the beat-to-beat intervals. Considering that for example the RMSSD range tends to lie between roughly 20–55 (Kim & Woo, 2011) and in our experience rarely exceeds 125, the error of 34.446 introduced by displacing 20% of peaks is more than large enough to bury any effects of external factors on HRV. Effects on BPM (and thus IBI as well) are negligible, however, reflecting their relative insensitivity to outliers.

3. Results

3.1. PPG data

The PPG dataset represents 20.7 h of PPG recordings split into 1240 one-minute segments. Due to sensor disconnects, 1.095 (18.25 h) of the data contained a heart rate signal. The signals were recorded at the tip of the finger as participants were driving in a driving simulator. The sensor placement did not interfere with driving. Participants were instructed to drive as they normally would in real life.

The data set was manually annotated by a human, then checked a second time to ensure accuracy. To annotate the set a custom tool was developed based on the Python's 'pyplot' plotting library. Each of the segments was visualised, and in the tool the annotator could manually mark peak locations, correct incorrectly placed peak locations and delete incorrect detections. Segments where little to no heart rate signal was present (for example due to the sensor detaching from the fingertip) were excluded. This left a total of 1.095 segments for the validation phase. A total of 89,837 peaks were detected by the algorithm. Of these, 84,845 (95.11%) were correctly accepted, and 2977 (3.34%) were correctly rejected automatically. This indicates that for 98.45% of all detections, the algorithm correctly labelled the peak locations. 957 (1.07%) peaks were incorrectly rejected. 426 (0.48%) peaks were incorrectly accepted. A total of 632 peaks were annotated as missed. Most of the incorrectly accepted peaks occur either because a peak location was marked not at but nearby its maximum (Fig. 2e) which induces a minor error, or because a diastolic (secondary) peak is marked as a peak (Fig. 2f) which induces a larger error. Future updates of the algorithm aim to further reduce these error rates.

We compared the performance of our algorithm with an implementation of the Pan-Tompkins QRS algorithm (Pan & Tompkins, 1985), as well as with an open source algorithm called HRVAS ECGViewer.[3] The latter was chosen because it is one of the first hits when searching for open source heart rate analysis software on Google and it shows high usage statistics. It is designed for Matlab, but a standalone version is also available. The Pan Tompkins algorithm is a computationally efficient algorithm widely used in ECG analysis.

The comparison results are displayed in Table 1. They indicate that our algorithm significantly outperforms the other two open source algorithms on PPG data. The peak position error is 0.89 (milliseconds), indicates that the mean of the errors between the actual peaks and the predicted peaks was low compared to the other two algorithms. The resulting peak-peak intervals were also more accurate compared to the other algorithms. This is likely due to less missed and less incorrectly accepted beats in our case (see Table 2).

Differences in BPM error are not very large. Since the BPM uses the mean of all peak-peak intervals in a segment, it is relatively robust to a few incorrectly placed peak positions. However, effects on heart rate variability measures are large. To evaluate HRV performance we selected the Standard Deviation of Successive Differences (SDSD), which is one often used

[3] See: https://github.com/jramshur/ECG_Viewer.

HRV measure that expresses how the intervals between the heart beats vary over time. It shows a large root mean squared error in the other two algorithms. This shows the importance of correctly identifying peak positions as well as identifying incorrectly labelled peaks, as deviations risk introducing substantial error to the output measures.

Table 2
Table showing how our algorithm compares to two other popular open source algorithms on key metrics.

Comparison of algorithm performance on PPG dataset (N = 1240)				
	Peak location error (ms)	RMSE peak-peak intervals	RMSE BPM	RMSE SDSD
Developed algorithm	0.89	29.64	3.77	167.77
Pan-Tompkins	7.62	267.67	10.73	1060.77
HRVAS ECGViewer	4.25	171.32	4.76	364.74

All analysis data, code and results are also available on the GitHub page (van Gent, 2017) in the form of Jupyter notebooks. These can be opened and viewed directly on GitHub, or downloaded and executed using the Python 3.6 Anaconda distribution.[4]

3.2. ECG data

The 1.000 sections selected from the ECG dataset comprise a total of 16.67 h of heart rate data. More information about the dataset is available in the publication of Jager et al. (2003).

The dataset was fully annotated. A total of 73.841 peaks were detected by the algorithm. Of these, 73.443 (99.46%) were correctly accepted and 190 (0.26%) were correctly rejected, representing a total of 99.72% of peaks correctly treated. 54 (0.07%) of peaks were incorrectly rejected, and 154 (0.21%) were incorrectly accepted. A total of 929 peaks were annotated as missed, meaning they were not detected by the algorithm. Note that with ECG, which has a more stable morphology, performance is significantly improved compared to PPG.

This performance was compared to the same algorithms as described above. The developed algorithm again showed superior performance on ECG data, although we were impressed with the performance of ECGViewer as well. The peak position error is lower compared to the PPG data, reflecting that the ECG waveform is easier to detect and more stable than the PPG waveform. The peak finding method we employ does not discriminate between the types of waveforms and can handle considerable morphological distortion.

The details are displayed in the table below.

Table 3
Table showing how our algorithm compares to two other popular open source algorithms on key metrics.

Comparison of algorithm performance on ECG dataset (N = 1000)				
	RMSE peak location	RMSE peak-peak intervals	RMSE BPM	RMSE SDSD
Developed algorithm	0.16	6.38	0.41	221.79
Pan-Tompkins	6.84	335.25	3.07	371.38
HRVAS ECGViewer	3.64	97.34	1.88	231.96

As with the PPG data, all analysis data, code and results are also available on the GitHub page (van Gent, 2017) in the form of Jupyter Notebooks. These can be used in conjunction with the manually annotated datasets to validate the performances on both datasets.

If must be noted that the lower performance of the other two algorithms is mainly due to a relatively small number of segments. Further analysis showed that for the developed algorithm, 98.7% of all segments had an error of 25 ms or less in the computed peak-peak intervals, with 58.1% showing no error at all. For the Pan-Tompkins implementation the majority (67.5%) also showed an error of 25 ms of less, with 9.5% showing no error at all. For the ECGViewer implementation 77% showed an error of less than 25 ms. No segments were without error. The reason for this is likely that this implementation uses a template matching system for beat detection, which while robust to noise, also creates slight errors in the positions of detected peaks because the template rarely matches the heartrate waveform in the measured signal perfectly.

Furthermore, the ECGViewer implementation for example failed to detect any peaks in 108 segments (10.8%), likely due to noise or deviating morphology. One such example is shown in the Fig. 3 below. These segments were excluded from the calculation of the performance measures from Table 3, so they do not negatively influence these measures.

[4] See: https://www.anaconda.com/download/.

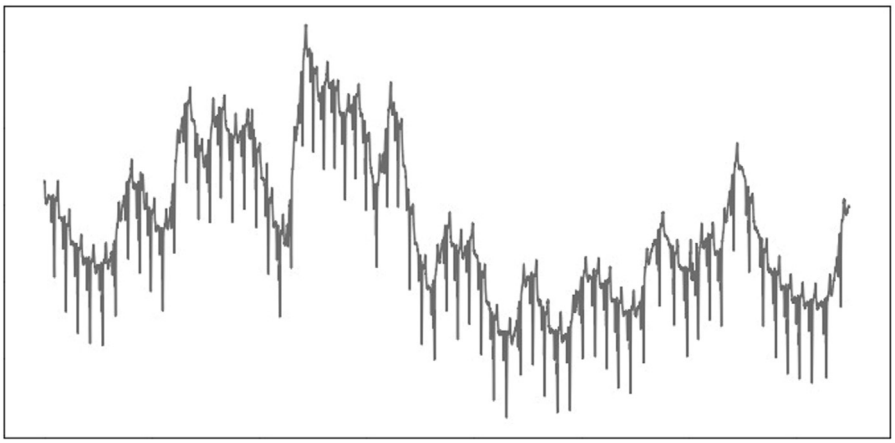

Fig. 6. An example of a noisy ECG recording where the ECGViewer implementation fails to detect any complexes. These segments were excluded from further analysis.

3.3. Additional information embedded in the heart rate signal: Breathing patterns

In addition to heart rate, the developed algorithm extracts breathing patterns from the collected heart rate data as well. Heart rate tends to increase during inhalation and decrease again during exhalation (Grossman & Taylor, 2007). This creates the possibility of extracting an estimate of the breathing rate from heart rate signals.

We've included a basic estimation method in the algorithm and validated it on an existing dataset (Karlen, Raman, Ansermino, & Dumont, 2013), that includes both PPG and respiratory data from patients undergoing surgery. Ground truth breathing rate was calculated from the CO_2 capnometry signal, which measures the increase in carbon dioxide concentration whenever the patient exhales. To extract breathing rate from the PPG signal the peak-peak intervals were upsampled, and their cycles marked. The following image visualises how the upsampled signal relates to the CO_2 capnometry signal.

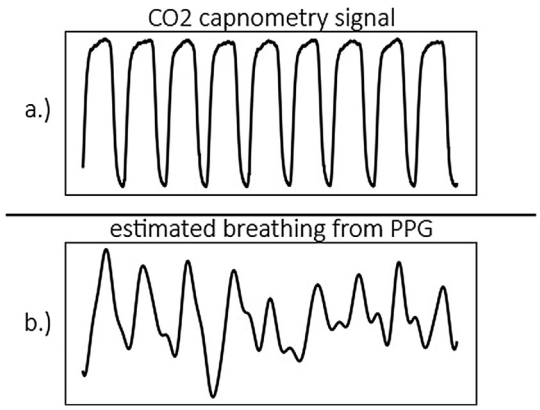

Fig. 7. Figure showing how the CO2 capnometry signal (top) relates to the breathing signal extracted from the PPG.

As discussed in (Grossman & Taylor, 2007), the relationship between heart rate variability and breathing rate is not a linear one and can be impacted by factors such as medication use and physical strain. The results of our validation on this dataset reflect this nonlinear relationship. The estimates correspond to the breathing rate determined by capnometry but include detection errors. See Table 4 for an overview of the detection errors expressed as the difference in Hertz between ground truth and calculated measures. The mean error present in the PPG estimation compared to the ground truth estimation of breathing rate, corresponds to a confidence interval of roughly 5–10%. This illustrates that the breathing rate extracted from the PPG signal should only be taken as an estimation and not an absolute value. Future improvements to the method might increase accuracy. The data files and analysis code are available on the GitHub page (van Gent, 2017).

Table 4
Table showing the difference between the ground truth breathing rate and the breathing rate estimated from the PPG.

	Mean	Median	Minimum	Maximum
Error magnitude	0.028	0.015	0	0.102

4. Discussion and conclusion

In this paper we have presented the validation analysis and results of a novel, robust heart rate analysis algorithm developed for use in lab settings, as well as in-vehicle and other mobile settings. The motivation to develop such an algorithm is that current available open source algorithms do not work well on noisy PPG data, are often highly technical or expensive to implement, and because low-cost commercial measurement devices offer no suitable solution for scientific purposes. The developed algorithm runs both in real-time and offline, on desktop computers and wearable (embedded) hardware. This makes it ideal for human factors studies seeking to incorporate heart rate analysis in their design, which includes studies into how automated vehicles can obtain and maintain an awareness of the driver's state.

We have evaluated the algorithm's performance on a manually annotated PPG and an ECG data set and compared the performance to two other available algorithms. Results showed superior performance on PPG data. This reflects that the lower performance of the two other algorithms is specific to the type of data: PPG data collected in the field using low-cost sensors has quite different signal and noise properties compared to ECG data, for which many available open source algorithms are designed. Despite the higher noise rate, PPG data can be collected using less intrusive or even contactless methods, which makes it ideal for real-world driving settings. On ECG data the differences between algorithms were less pronounced.

Making automated vehicles smarter means they need to be aware of the driver's state. Heart rate is one physiological marker that allows the estimation of driver state on several levels. By offering an openly available and validated toolkit for heart rate analysis, we aim to increase the research possibilities into this field, as well as the reliability and reproducibility of results obtained.

One limitation of the present validation is that it was performed on 2.095 one-minute segments, due to the time-intensive nature of the manual annotation. The annotation was done by hand, which means minor errors can exist in the annotations. Although we believe that the reported performance is a good reflection of real-world performance due to the two different datasets used, a larger validation will add further confidence to the results. Furthermore, the included breathing rate estimation should be taken as an estimated value rather than an absolute one, since the relationship between heart rate and breathing rate is not linear. Future steps include this larger validation, as well as increasing the accuracy and functionality of the algorithm, and comparing its performance to commercially available solutions (see Figs. 5–7).

Funding statement

Part of the software has been developed within the "Taking the Fast Lane" project, funded by NWO TTW, project number 13771.

References

Bernardi, L., Leuzzi, S., Radaelli, A., Passino, C., Johnston, J. A., & Sleight, P. (1994). Low-frequency spontaneous fluctuations of R-R interval and blood pressure in conscious humans: a baroreceptor or central phenomenon? *Clinical Science (London, England : 1979), 87*(6), 649–654.

Birrel, S., Young, M., Stanton, N., & Jennings, P. (2017). Using adaptive interfaces to encourage smart driving and their effect on driver workload, 484, 764. https://doi.org/10.1007/978-3-319-41682-3.

Brookhuis, K. A., & de Waard, D. (2010). Monitoring drivers' mental workload in driving simulators using physiological measures. *Accident Analysis and Prevention, 42*(3), 898–903. https://doi.org/10.1016/j.aap.2009.06.001.

Brouwer, A., & Dijksterhuis, C. (2015). Physiological correlates of mental effort as manipulated through lane width during simulated driving. In Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction (ACII) (pp. 42–48).

Danisman, T., Bilasco, I. M., Djeraba, C., & Ihaddadene, N. (2010). Drowsy driver detection system using eye blink patterns. In 2010 International Conference on Machine and Web Intelligence, ICMWI 2010 - Proceedings (pp. 230–233). https://doi.org/10.1109/ICMWI.2010.5648121.

Elgendi, M. (2012). On the analysis of fingertip photoplethysmogram signals. *Current Cardiology Reviews, 8*(1), 14–25. https://doi.org/10.2174/157340312801215782.

Fallahi, M., Motamedzade, M., Heidarimoghadam, R., Soltanian, A. R., & Miyake, S. (2016). Effects of mental workload on physiological and subjective responses during traffic density monitoring: A field study. *Applied Ergonomics, 52*, 95–103. https://doi.org/10.1016/j.apergo.2015.07.009.

Farah, H., Koutsopoulos, H. N., Saifuzzaman, M., Kölbl, R., Fuchs, S., & Bankosegger, D. (2012). Evaluation of the effect of cooperative infrastructure-to-vehicle systems on driver behavior. *Transportation Research Part C: Emerging Technologies, 21*(1), 42–56. https://doi.org/10.1016/j.trc.2011.08.006.

Grossman, P., & Taylor, E. W. (2007). Toward understanding respiratory sinus arrhythmia: Relations to cardiac vagal tone, evolution and biobehavioral functions. *Biological Psychology, 74*(2), 263–285. https://doi.org/10.1016/j.biopsycho.2005.11.014.

Healey, J. A., & Picard, R. W. (2005). Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems, 6*(2), 156–166.

Hoogendoorn, R., van Arem, B., & Hoogendoorn, S. (2014). Automated driving, traffic flow efficiency, and human factors. *Transportation Research Record: Journal of the Transportation Research Board, 2422*(2422), 113–120. https://doi.org/10.3141/2422-13.

Jager, F., Taddei, A., Moody, G. B., Emdin, M., Antolič, G., Dorn, R., ... Mark, R. G. (2003). Long-term ST database: A reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia. *Medical and Biological Engineering and Computing, 41*(2), 172–182. https://doi.org/10.1007/BF02344885.

Jamson, A. H., Merat, N., Carsten, O., & Lai, F. (2011). Fully-automated driving: the road to future vehicles. In Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design (pp. 2–9). https://doi.org/10.17077/drivingassessment.1370.

Jarvis, J., Putze, F., Heger, D., & Schultz, T. (2011). Multimodal person independent recognition of workload related biosignal patterns. In Proceedings of the 13th International Conference on Multimodal Interfaces - ICMI '11, 205. https://doi.org/10.1145/2070481.2070516.

Karlen, W., Raman, S., Ansermino, J. M., & Dumont, G. A. (2013). Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE Transactions on Biomedical Engineering, 60*(7), 1946–1953. https://doi.org/10.1109/TBME.2013.2246160.

Kim, G. M., & Woo, J. M. (2011). Determinants for heart rate variability in a normal Korean population. *Journal of Korean Medical Science, 26*(10), 1293–1298. https://doi.org/10.3346/jkms.2011.26.10.1293.

Kim, S., Chun, J., & Dey, A. K. (2015). Sensors Know When to Interrupt You in the Car. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15 (pp. 487–496). https://doi.org/10.1145/2702123.2702409.

Kirst, M., Glauner, B., & Ottenbacher, J. (2011). Using DWT for ECG motion artifact reduction with noise-correlating signals. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS (pp. 4804–4807). https://doi.org/10.1109/IEMBS.2011.6091190.

Liang, Y., Reyes, M. L., & Lee, J. D. (2007). Real-time detection of driver cognitive distraction using support vector machines. *IEEE Transactions on Intelligent Transportation Systems, 8*(2), 340–350. https://doi.org/10.1109/TITS.2007.895298.

Mehler, B., Reimer, B., & Coughlin, J. F. (2012). Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: An on-road study across three age groups. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 54*(3), 396–412. https://doi.org/10.1177/0018720812442086.

Mehler, B, Reimer, B., & Wang, Y. (2011). Comparison of heart rate and heart rate variability indices in distinguishing single task driving and driving under secondary cognitive workload. In Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design (pp. 590–597).

Mehler, B., Reimer, B., Coughlin, J. F., & Dusek, J. A. (2010). Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers. *Transportation Research Record: Journal of the Transportation Research Board, 2138*(1), 6–12. https://doi.org/10.3141/2138-02.

Merat, N., Jamson, A. H., Lai, F. C. H., Daly, M., & Carsten, O. M. J. (2014). Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation Research Part F: Traffic Psychology and Behaviour, 27*(PB), 274–282. https://doi.org/10.1016/j.trf.2014.09.005.

Millasseau, S. C., Guigui, F. G., Kelly, R. P., Prasad, K., Cockcroft, J. R., Ritter, J. M., & Chowienczyk, P. J. (2000). Noninvasive assessment of the digital volume pulse: comparison with the peripheral noninvasive assessment of the digital volume pulse comparison with the peripheral pressure pulse. *Hypertension, 36*, 952–956. https://doi.org/10.1161/01.HYP.36.6.952.

Miyaji, M., Danno, M., Kawanaka, H., & Oguri, K. (2008). Driver's cognitive distraction detection using adaboost on pattern recognition basis. In Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2008 (pp. 51–56). https://doi.org/10.1109/ICVES.2008.4640853.

Montano, N., Porta, A., Cogliati, C., Costantino, G., Tobaldini, E., Casali, K. R., & Iellamo, F. (2009). Heart rate variability explored in the frequency domain: A tool to investigate the link between heart and behavior. *Neuroscience and Biobehavioral Reviews, 33*(2), 71–80. https://doi.org/10.1016/j.neubiorev.2008.07.006.

Pan, J., & Tompkins, W. J. (1985). A simple real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering, BME-32*(3), 230–236. https://doi.org/10.1109/IEMBS.1996.647473.

Park, H. S., & Kim, K. H. (2015). Adaptive multimodal in-vehicle information system for safe driving. *ETRI Journal, 37*(3), 626–636. https://doi.org/10.4218/etrij.15.0114.1104.

Reimer, B., Mehler, B., & Coughlin, J. F. (2016). Reductions in self-reported stress and anticipatory heart rate with the use of a semi-automated parallel parking system. *Applied Ergonomics, 52*, 120–127. https://doi.org/10.1016/j.apergo.2015.07.008.

Reimer, B., Mehler, B., Coughlin, J. F., Roy, N., & Dusek, J. A. (2011). The impact of a naturalistic hands-free cellular phone task on heart rate and simulated driving performance in two age groups. *Transportation Research Part F: Traffic Psychology and Behaviour, 14*(1), 13–25. https://doi.org/10.1016/j.trf.2010.09.002.

Selvaraj, N., Jaryal, A., Santhosh, J., Deepak, K. K., & Anand, S. (2008). Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of Medical Engineering and Technology, 32*(6), 479–484. https://doi.org/10.1080/03091900701781317.

Solovey, E. T., Zec, M., Garcia Perez, E. A., Reimer, B., & Mehler, B. (2014). Classifying driver workload using physiological and driving performance data. In Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14 (pp. 4057–4066). https://doi.org/10.1145/2556288.2557068.

Stapel, J., Mullakkal-Babu, F. A., & Happee, R. (2017). Driver behaviour and workload in an on-road automated vehicle. In Proceedings of the RSS2017 Conference.

Sun, Y., Hu, S., Azorin-Peris, V., Kalawsky, R., & Greenwald, S. (2012). Noncontact imaging photoplethysmography to effectively access pulse rate variability. *Journal of Biomedical Optics, 18*(6). https://doi.org/10.1117/1.JBO.18.6.061205 061205.

van Gent, P. (2017). Python heart rate analysis toolkit. Retrieved from https://github.com/paulvangentcom/heartrate_analysis_python.

van Gent, P. (2018). Arduino heart rate analysis toolkit. Retrieved from https://github.com/paulvangentcom/heartrate_analysis_Arduino.

van Gent, P., Farah, H., van Nes, N., & van Arem, B. (2017). Towards real-time, nonintrusive estimation of driver workload: A simulator study. In Road Safety and Simulation 2017 Conference Proceedings.

van Gent, P., Farah, H., van Nes, N., & van Arem, B. (n Gent et al., submitted for publication). A conceptual model for persuasive in-vehicle technology to influence tactical level driver behaviour. *Transportation Research Part F*, 1–20. submitted for publication.

van Gent, P., Farah, H., van Nes, N., & van Arem, B. (n Gent et al., submitted for publication). Analysing noisy driver physiology real-time using off-the-shelf sensors: Heart rate analysis software from the taking the fast lane project. *Journal of Open Research Software*. submitted for publication.

van Gent, P., Melman, T., Farah, H., van Nes, N., & van Arem, B. (2018). Multi-level driver workload prediction using machine learning and off-the-shelf sensors. *Transportation Research Record: Journal of the Transportation Research Board*.. https://doi.org/10.1177/0361198118790372.

Zhai, J., & Barreto, A. (2006). Stress Recognition Using Non-invasive Technology. In Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS) (pp. 395–400).