Biomedical Signal Processing and Control 101 (2025) 107158

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Enhancing rPPG pulse-signal recovery by facial sampling and PSD Clustering

Giuseppe Boccignone [a], Donatello Conte [b], Vittorio Cuculo [c], Alessandro D'Amelio [a], Giuliano Grossi [a,*], Raffaella Lanzarotti [a]

[a] Dipartimento di Informatica, Università degli Studi di Milano, Via Celoria 18, Milano, 20133, Italy
[b] Laboratoire d'Informatique Fondamentale et Appliquée (LIFAT - EA 6300), Université de Tours, Ave J. Portalis, Tours, 37000, France
[c] Dipartimento di Ingegneria "Enzo Ferrari", Università degli Studi di Modena e Reggio Emilia, Via Università 4, Modena, 41121, Italy

ARTICLE INFO

Dataset link: github.com/phuselab/pyVHR

Keywords:
Remote photoplethysmography (rPPG)
Heart rate variability
Pulse rate estimation
PSD analysis
Face mesh landmarks
Unsupervised circle clustering
Nonlinear dynamical systems

ABSTRACT

In order to increase the accuracy of traditional methods for camera-based pulse rate estimation, we propose a novel systemic approach that extracts multiple remote photoplethysmography (rPPG) signals from a set of scattered facial patches and effectively separates good estimates from noisy ones via a novel unsupervised Power Spectral Density (PSD) clustering method. In contrast to commonly adopted rPPG pipelines, which are often challenged by rigid head movements, facial expressions, and rapidly changing lighting conditions, our patch-oriented solution leverages the key feature of patch recurrence in video sequences. Instead of focusing on a small group of specific Regions of Interest (ROIs), our method adaptively selects a set of patches tracked across successive frames. The spatio-temporal self-similarity among these patches provides powerful internal statistics that significantly enhance standard techniques for rPPG assessment. Our main contribution is a novel unsupervised discriminatory strategy called CircleClustering, which naturally separates PSDs into those with low intra-class variability from those with high intra- and inter-class inhomogeneity. Extensive experimental results demonstrate the overall superiority of our patch-based clustering method compared to both traditional signal processing-based rPPG techniques and recent supervised deep learning-based models for rPPG recovery.

1. Introduction

Remote photoplethysmography (rPPG) enables contactless Heart Rate (HR) monitoring by capturing the subtle skin color variations induced by blood volume pulse, through a video camera [1]. This technology has recently seen increased interest due to the wide variety of its downstream applications, such as sleep monitoring [2], telemedicine [3], emotion recognition [4], estimation of other physiological signals [5] or DeepFake detection [6,7].

The rationale behind rPPG is simple. According to the dichromatic reflection model (cf. Fig. 1), the camera signal can be decomposed into specular and diffuse reflections [8]: the former is a mirror-like reflection that does not deliver any significant cardiac detail; conversely, diffuse reflection originates from light that penetrates through the layers of the skin and diffuses back. As a result, the latter represents a better carrier of physiological information.

In such an effort, the vast majority of rPPG methods exploit either signal processing or deep learning techniques to effectively extract diffuse reflections from RGB traces. Clearly, their effectiveness strongly depends on the reliability of the considered Regions of Interest (ROIs). In principle, it is paramount to select ROIs that are not dominated by specular reflection.

Classic approaches either rely on the detection of specific ROIs (*e.g.* forehead or cheeks) or - by exploiting the spatial redundancy of a color camera sensor - obtain RGB traces by averaging the whole face skin color intensities (this is sometimes referred to as *holistic* approach [9,10]). Notoriously, these approaches suffer from lack of robustness due to various reasons, such as sudden changes in pixel intensity values caused by facial expressions, uneven or varying face illumination, and motion-induced color variations.

More recently, end-to-end deep-learning techniques have been adopted; these typically ingest lightly pre-processed face videos and defer the selection of the appropriate ROI to the learning algorithm. This can be achieved, for example, via neural attention mechanisms [11,12]. Despite the undeniable advantages, deep learning-based solutions exhibit some downsides related to the need of huge amounts of data, a cogently critical condition to be satisfied in the rPPG field; indeed, building rPPG corpora requires the measurement of a physiological ground truth (necessitating contact sensors) and the recording of uncompressed (or lightly compressed) videos [13].

* Corresponding author.
E-mail addresses: giuseppe.boccignone@unimi.it (G. Boccignone), donatello.conte@univ-tours.fr (D. Conte), vittorio.cuculo@unimore.it (V. Cuculo), alessandro.damelio@unimi.it (A. D'Amelio), giuliano.grossi@unimi.it (G. Grossi), raffaella.lanzarotti@unimi.it (R. Lanzarotti).

https://doi.org/10.1016/j.bspc.2024.107158
Received 7 February 2024; Received in revised form 8 October 2024; Accepted 2 November 2024
Available online 19 November 2024

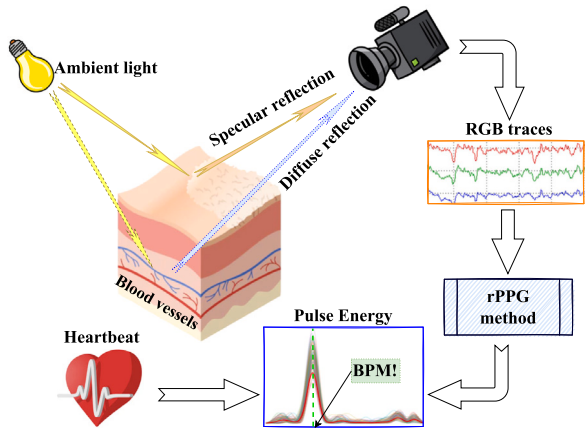


Fig. 1. Dichromatic reflection model: diffusion reflection is related to pulse signal and any rPPG method uses signal processing to extract heartbeat information.

With the aim of mitigating such problems, in this work, we present a novel approach that leverages the spatio-temporal similarities existing in the diverse set of ROIs scattered on a face video. This is accomplished by a novel unsupervised algorithm (CircleClustering) able to adaptively and effectively select the most useful ROIs without requiring training or ground truth physiological signals.

The approach is based on the fact that face images contain local information that is relatively coherent across certain patches(particularly those conveying more diffuse reflection) while other patches tend to show more noisy, incoherent patterns. The overall objective is to exploit such substantial internal information redundancy, eventually yielding reliable internal statistics [14]. More specifically, many small patches (e.g. 10×10 pixels) are likely to present similar spectral characteristics as representing different measurements of the same underlying process (cardiac activity). Similar properties have been extensively exploited by classical image processing algorithms that tackle problems such as denoising, super-resolution and text synthesis [14]. A common ground of these approaches is to split the image into possibly overlapping patches (ROIs) and arrange them in a suitable way according to their similarity to gain predictive performance.

A fundamental difficulty when comparing two or more patches in a real scenario is deciding whether the differences should be ascribed to either noise or intrinsic dissimilarity. The very same problem applies to the rPPG context that presents challenges related to uncontrolled illumination, different skin-tones, and significant body motions. Moreover, it is worth noticing that as to the rPPG problem, local patches are not represented by standard "low-level" features, but are processed through complex models, the actual rPPG technique employed. This inevitably entails that classical noise models (*e.g.* Gaussian noise) are not applicable here.

In this work, we propose a novel clustering algorithm that acts on the spectrum of locally estimated rPPG signals, referred to as CircleClustering. Empirical results demonstrate that this algorithm effectively identifies two well-separated sets of ROIs, defining a "good" cluster and a "bad" one. More specifically, it provides a bipartition with the following asymmetric property: the good cluster groups together homogeneous Power Spectral Densities (PSDs) with peaks concentrated around the same frequency, while the bad cluster collects PSDs that are inhomogeneous with one another (intra-class variability) and also dissimilar from those in the good cluster (inter-class variability). From a theoretical perspective, CircleClustering relies on dynamical system theory, a principled tool for modeling and studying phenomena that undergo spatial and temporal evolution. In brief, the algorithm associates PSD points lying in a Euclidean space of large dimensions (of the order of a few hundred) with elements on the unit circle, assumed by scalar variables representing angles in the real interval of size 2π. Given an arbitrary metric to measure distance or similarity between PSDs, a nonlinear autonomous dynamical system with state on the edge hypercube 2π is activated to minimize an energy (cost) function; the cost combines such distances and the scalar variables instantiated on the hypercube. An asynchronous dynamics is applied to lead the system to a point arbitrarily close to a fixed point that is asymptotically stable in a finite number of steps. The system is shown to always converge and the equilibrium point found represents a local minimum of the cost function, which in turn represents a Lyapunov function for the dynamical system [15]. At the end of this process, the best separating hyperplane of the points on the unit circle is found through a simple greedy technique, from which a bipartition of the original PSD set is automatically derived due to the isomorphism with the points on the circle. A final pulse-rate estimation for each patch group is achieved by Gaussian fitting the average PSDs of the two clusters and selecting the cluster with the best fit.

In a crude summary, the relevant and innovative points of our approach can be summarized as follows.

- A novel adaptive patch-oriented approach for better capturing the faint presence of blood pulsations from a set very noisy rPPG estimates.
- A novel two-stage PSD clustering method able to group PSDs exhibiting high spectral coherence (good cluster) from dissimilar or highly inconsistent ones (bad cluster).

We perform extensive experiments on multiple datasets and test several classic rPPG methods with or without the proposed CircleClustering algorithm. Results demonstrate the effectiveness of the proposed method, even compared to state-of-the-art deep learning-based approaches.

The paper unfolds as follows. Section 1.1 summarizes previous rPPG-related research focusing specifically on the ROI selection problem. Section 2.1 describes the adopted face sampling technique. Section 2.2 details the spectral analysis performed at the patch level of the rPPG measurements used in this study. In Section 2.3 the CircleClustering algorithm is proposed and analyzed. Section 3 reports experimental work, demonstrating the ability of the proposed method to infer the right pulse rate with minimal error. Eventually, conclusions are drawn in Section 4.

1.1. Related works

A great deal of research work witnesses the crucial role of ROI selection as a fundamental first step of many rPPG techniques to obtain reliable pulse signals. Main reasons are to be attributed to a number of factors ranging from anatomical elements to a wide range of noise types. In [16] it is argued that a critical element is the nonuniform thickness of the skin in all areas of the face, which is the reason why one cannot obtain the same diffuse reflection information in each zone. Similarly [17] states that some skin regions contain more rPPG signal than others, mainly for physiological reasons, suggesting explicitly favoring areas where information is more predominant using a spatially weighted average of skin pixels based on a trained model. Instead, the authors of [18] propose dynamically selecting regions that perform block-based spatio-temporal division and final clustering to find adaptive ROIs driven by SNR.

In recent years, the literature has been driven by the flourishing of deep learning (DL)-based approaches (for recent reviews, see [19–21]). Some DL methods consider hybrid approaches where DL is used to cover only some steps over the pipeline. In this group, several papers highlight the necessity to extract ROIs in order to only process the most informative regions. For example, in [22] the ROI corresponding to the central part of the face (including the cheeks and nose) is detected and fed as input into PRnet. In [23] the ROIs corresponding to the forehead and to the cheeks are considered as input to a Siamese-rPPG Network.

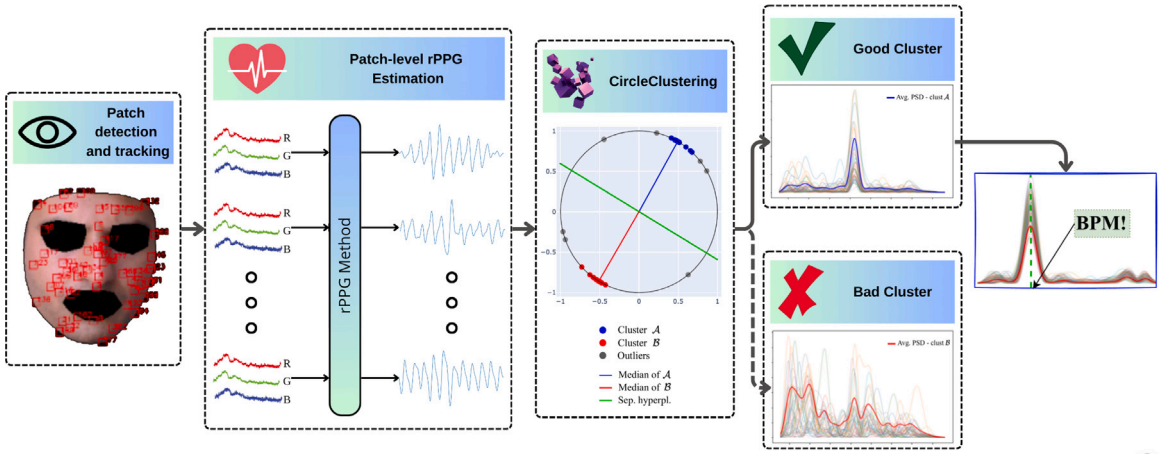


Fig. 2. Pipeline summarizing the approach for adaptive ROIs (patches) selection via the proposed CircleClustering algorithm to enhance camera-based heart rate estimation via rPPG.

All these approaches have certain limitations. On the one hand, the static selection of ROIs, while efficient, lacks flexibility to adapt to varying subjects and environmental conditions. On the other hand, DL-based methods, while adaptable and powerful, have several drawbacks. First, they require large amounts of labeled data, which is often scarce in rPPG applications. Furthermore, these models can overfit noisy data and may be sensitive to factors such as lighting conditions, facial movements, and environmental changes, which can compromise their robustness. Moreover, DL models are frequently regarded as "black boxes" making their decision-making processes difficult to interpret, which is a critical concern in health monitoring. Finally, these models require significant computational resources, including GPUs and large datasets for effective training.

2. The method

In this section, we provide a thorough description of the proposed approach, which involves the detection and tracking of ROIs (face patches), rPPG estimation, and the novel PSD clustering technique that adaptively and automatically selects the most useful ROIs. The pipeline illustrating the overall proposal is shown in Fig. 2.

2.1. Patch detection and tracking

Following the typical rPPG-based pulse rates estimation pipeline, we assume a sequence of T (windowed) RGB frames as input. The tth frame (for $t = 1, \dots, T$) represents the collection of pixels given by the vectors $\mathbf{c}_{i,j}(t) = (r_{i,j}(t), g_{i,j}(t), b_{i,j}(t))^\top$, where $r_{i,j}(t), g_{i,j}(t), b_{i,j}(t)$ represent the red, green and blue channels for the pixel in position (i, j), respectively, and $^\top$ denotes vector transposition.

Unless stated otherwise, in the rest of the paper, we implicitly assume that a given video $v \in \mathbb{R}^{h \times w \times 3 \times N}$ of N RGB frames of size (h, w) is sliced into $K = \lfloor N/(T - S) \rfloor$ overlapping windows, where T is the number of frames per window, and S the stride. Therefore, each pulse rate estimate derived from this modeling is limited to a window of T frames and is conducted independently of the others. Based on this, the whole evaluation process takes as input a video and produces K BPM estimates, one for each considered window.

In the patch-based approach, we consider a collection of scattered patches centered on $\mathcal{L} = [1..L]$ landmarks from which a bunch of RGB traces can be extracted (see the forthcoming section for details). Examples of different sets of face landmarks are given in Fig. 3. Fig. 3(a) shows a set of landmark locations sampled from a quasi-uniform spatial distribution. The subsequent figures (b-d) depict different sets of patches (each centered on a landmark) with increasing spatial density (from 25 to 100). As can be seen from the figure, non-skin pixels (e.g., eyes, eyebrows, mouth, and background) are excluded from the patch sampling process, as these are typically considered noisy areas for the rPPG estimation process.

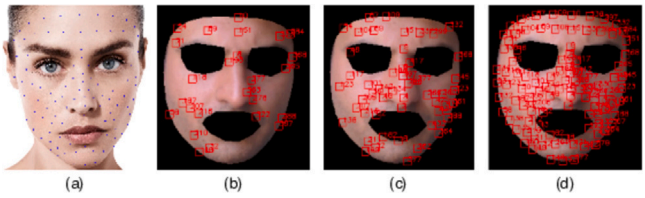


Fig. 3. Model with 100 quasi-uniformly spread face mesh landmarks (a). Patches centered on 25 (b), 50 (c), and 100 (d) landmarks respectively.

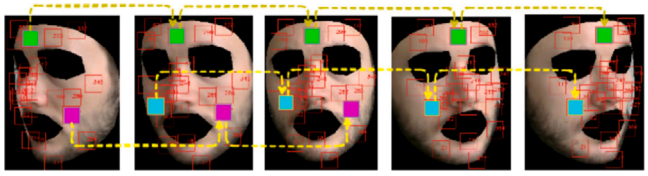


Fig. 4. Landmarks automatically tracked by MediaPipe and correspondent patch tracking.

An example of landmark extraction and tracking is illustrated in Fig. 4, where three patches were selected for visualization from the forehead, left cheek, and right cheek areas. Note that a patch may disappear due to the subject's movement, thus providing partial or no contribution. Typically, a fairly high number of patches is chosen to better deal with the many variations in boundary conditions (movement, ambient light, etc.) and to achieve higher confidence levels in the subsequent inferences.

2.1.1. Filtering of RGB color signals

The second processing step is ubiquitous in model-based rPPG methods and is related to spatial quantization. Given an input video that contains a face, we spatially average the RGB values of the pixels within the sampled patches in each frame. These values are then temporally concatenated to yield RGB color traces.

Specifically, given a mesh of scattered patches centered on landmarks $\mathcal{L}$, for each landmark $l \in \mathcal{L}$ we select a patch P_l centered in l. For each frame $t \in [1..T]$ we compute the average color intensities over $P_l(t)$ (RGB color trace):

$$\bar{\mathbf{c}}_l(t) = (\bar{r}_l(t), \bar{g}_l(t), \bar{b}_l(t))^\top = \frac{1}{|P_l(t)|} \sum_{(i,j) \in P_l(t)} \mathbf{c}_{i,j}(t) \quad (1)$$

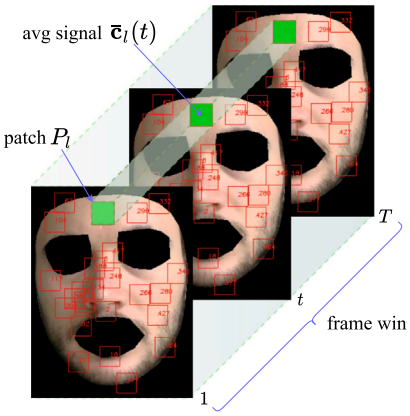


Fig. 5. Patch tracking within a frame temporal window, and the RGB color signal computation at frame t.

where $|P_l(t)|$ is the number of pixels in $P_l(t)$.

Fig. 5 shows how the patch-based split and tracking procedure is implemented along with the RGB color signal $\bar{\mathbf{c}}_l(t)$ collected for each frame t on the patch $P_l(t)$.

Assuming that a video has T frames, the RGB traces $\bar{\mathbf{c}}_l$ provide a vector in the $\mathbb{R}^{3\times T}$ space. For a video-camera recording at a rate of τ frames-per-second (fps), the time span of the data in Eq. (1) covers T/τ seconds.

It is good practice to eliminate the DC-colors[1] by temporally normalizing each row of (1) as:

$$\tilde{\mathbf{c}}_l = \left[\frac{\bar{\mathbf{r}}_l}{\mu(\bar{\mathbf{r}}_l)}, \frac{\bar{\mathbf{g}}_l}{\mu(\bar{\mathbf{g}}_l)}, \frac{\bar{\mathbf{b}}_l}{\mu(\bar{\mathbf{b}}_l)}\right]^\top \in \mathbb{R}^{3\times T},$$

where the temporal average operator $\mu(\cdot)$ is applied to each RGB channel [24]. Both spatial and temporal normalization play an important role in rPPG estimation: spatial pixel averaging breaks down the camera quantization error [8], while temporal normalization aims to eliminate the dependency of $\mathbf{c}_l$ on the average skin reflection color, considered as the large steady component over a time interval.

In the pre-processing stage, common filtering procedures are often applied to retain only the frequencies within the human heart rate range (40–240 BPM, corresponding to 0.65–4 Hz), as this has a significant impact on pulse extraction. Temporal filtering techniques are typically employed, such as detrending, moving-average, and bandpass filters. Detrending helps in isolating the pulsatile component of rPPG by drastically reducing the low frequencies of the raw signal which determines non-stationary trends of signals [25,26]. The moving-average filter smooths the signal by suppressing high-frequency random noise caused by sudden color changes due to light or motions using the temporal average of consecutive frames. Classical filters such as the bandpass filter are also used to remove irrelevant frequencies outside the heart rate bandwidth (e.g., Butterworth filters).

All the aforementioned filters are often used in combination [25], and the resulting signal obtained by applying k filters in cascade to spatially and temporally normalized signals $\tilde{\mathbf{c}}_l$ related to the patch P_l thus becomes:

$$\hat{\mathbf{c}}_l = \mathrm{FILT}_1(\cdots \mathrm{FILT}_k(\tilde{\mathbf{c}}_l)\cdots).$$

Fig. 6 shows some samples of raw RGB color signals (top picture) obtained from patches (randomly choosing only one channel among $\hat{\mathbf{r}}_l$, $\hat{\mathbf{g}}_l$ and $\hat{\mathbf{b}}_l$), together with their filtered versions (bottom picture) after the application of detrending and bandpass filters.

[1] Here the DC-colors refer to the temporally averaged colors of skin and background, where it is assumed that such averages are quite stable over a short period.

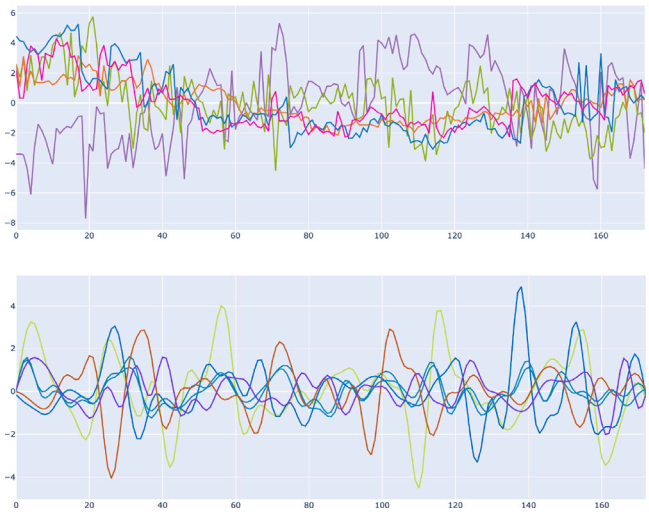


Fig. 6. Samples taken from some patches of both raw (top) and filtered (bottom) RGB color signals after the application of detrending and bandpass. Line colors are purely random, used to distinguish the overlapped plots.

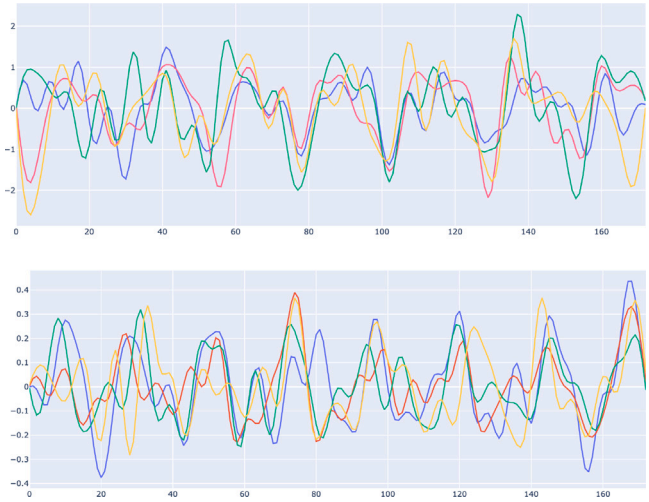


Fig. 7. rPPG signals provided by the two methods CHROM (upper) and LGI (lower) respectively, after RGB trace extraction from various patches.

An example of cardiac signals recovered by two commonly adopted rPPG methods is shown in Fig. 7 (see Section 3 for details).

2.2. Patch-level spectral analysis

From a pure signal processing point of view, rPPG measurement poses two relevant issues: the periodicity of the underlying phenomenon and the random effects introduced by noise. By and large, most rPPG pipelines analyze the frequency domain content of the estimated signals as opposed to a time domain analysis. A reason for supporting this approach is that the noise spectrum has almost certainly a different spectral line, helping to discriminate the most informative frequency peaks. Under such circumstances, spectral analysis is performed via Power Spectral Density (PSD) estimation. PSD provides information on the power distribution as a function of frequency, provided that the signal is at least weakly stationary to avoid distortions in the time and frequency domains. In order to grant a weaker form of stationarity, here we compute the PSD on small intervals, e.g. $5 \div 10$ seconds, so as to preserve the significant peaks in the pulse frequency band ($[40, 240]$ BPM). Here, the PSD

is computed via the discrete time Fourier transform (DFT) using the Welch's method, which employs both averaging and smoothing to analyze the underlying random phenomenon.

Formally, given an rPPG signal b_l of length T for each patch $l \in \mathcal{L}$, the sequence $b_l = (x(1), \dots, x(T))$ is divided into S segments (or windows) of length M, with a shift of K samples between adjacent segments, thus producing an overlap of $M - K$ samples. Denote a segment

$$x_j(t) = x((j-1)K + t), \quad t = 1, \dots, M, \quad j = 1, \dots, S$$

where $(j-1)K$ is the starting element of the segment j. As proposed by Welch method, the jth segment gives rise to the periodogram

$$S_j^l(\omega) = \frac{1}{M\Phi} \left| \sum_{t=1}^{M} w(t) x_j(t) e^{-i\omega t} \right|^2,$$

where $w(t)$ is a smoothing temporal window (typically of Hamming or Hanning) and $\Phi = \sum_t w(t)^2$ denotes the power spectral density of the window. Assuming 50% overlap, then $K = M/2$, from which $S \approx 2T/M$.

Welch's method can be efficiently computed via FFT, and it is one of the most frequently used methods for PSD estimation obtained by averaging the periodograms of the individual overlapping windows:

$$S^l(\omega) = \frac{1}{S} \sum_{j=1}^{S} S_j^l(\omega). \tag{2}$$

For a periodic signal such as rPPG pulse signals, the power is concentrated in extremely narrow bands of frequencies. Hence, finding the estimate of the BPM value given by the rPPG $\mathbf{b}_l$ signal of the patch l, leads us to search for the maximum peak of the associated PSD $S^l(\omega)$, whose most representative patterns generally show a unique lobe centered on the frequency correlated with the pulse (some examples are shown in Fig. 8).

Denoting by $\tau = 1/\text{fps}$ the time elapsed between two frames and assuming that a patch is analyzed for T frames starting from time t_0, the actual sampling times correspond to the sequence $t_0 + n\tau$, with $n = 0, 1, \dots, T-1$, for a time window of $T\tau$ seconds. The frequency $f = \omega/2\pi$ (expressed in Hz) falls in the range $(-1/2\tau + 1/Q\tau)$ to $1/2\tau$ Hz, with a resolution $\nu = 1/Q\tau$ Hz when using the FFT from (2) with exactly Q samples. The peak, being generally unique, is therefore easily obtained in the set of frequencies $\Omega = \{-1/2\tau + k\nu : k = 1, \dots, Q\}$ as

$$f^l = \underset{f_k \in \Omega}{\text{argmax}} \left\{ S^l(f_k) \right\}. \tag{3}$$

Note that the final frequency should be expressed in BPM (Beats per Minute): $f^l_{\text{BPM}} = 60 \times f^l$.

As in general the video frame rate is rather low (normally fps = 25 or fps = 30 for standard video) it is useful to define a higher frequency resolution by setting, for example, $Q = 2048$, which is a reasonable compromise for video segments shorter than 10 s.

2.3. PSD clustering

In this section, we introduce a novel two-stage clustering method to separate the PSD set $\mathcal{S} = \{S^l(\omega) \in \mathbb{R}^T : l \in \mathcal{L}\}$ into two clusters, denoted $\mathcal{A}$ and $\mathcal{B}$, one exhibiting high spectral coherence (good cluster) and one presenting more dissimilar or highly incoherent PSDs (bad cluster). This transforms the problem of final BPM estimation into choosing the cluster that collects the PSD peaks f^l_{BPM} with high internal consistency. For example, in Fig. 8 it can be seen that the two outer PSDs are much more similar to each other than the middle one is to the others.

Define for each pair $S_i, S_j \in \mathcal{S}$ (we remove the independent variable ω for simplicity), a pairwise distance matrix $W = (w_{ij})$, of size $L \times L$. The element $w_{ij} = \text{distance}(S_i, S_j)$ gives the degree of (dis)similarity between the two vectors S_i and S_j in an inner product space. In our

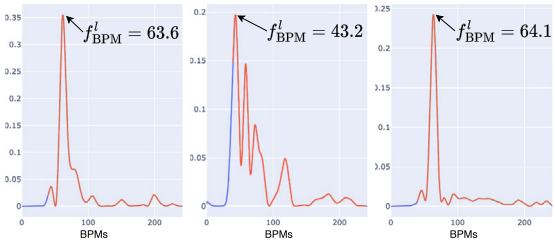


Fig. 8. PSD examples extracted from three patches of the same rPPG segment and related peaks in BPMs highlighted.

experiments, the cosine distance provides the most effective results, giving a degree of dissimilarity between 0 and 1. For practical reasons that will become clear later, we associate with each vector $S_i \in \mathcal{S}$ a variable assuming values on the unit circle $\xi_i \in \mathbb{S}_2$. The main advantage of this trick is to make the optimization of the functional simpler, as the original vectors S_i, typically lie in a high-dimensional space. A second non-negligible advantage comes from the fact that, since the inner product between two vectors $\xi_i = (\cos\theta_i, \sin\theta_i)$ and $\xi_j = (\cos\theta_j, \sin\theta_j)$ verifies $\xi_i \cdot \xi_j = \cos(\theta_i - \theta_j)$, the relative similarity of two vectors in $\mathbb{S}_2$ depends on the scalar variables θ_i and θ_j and can be written as a function that (with some abuse of notation) assumes the form:

$$d(\xi_i, \xi_j) = \frac{1 - \xi_i \cdot \xi_j}{2} = d(\theta_i, \theta_j) = \frac{1 - \cos(\theta_i - \theta_j)}{2}. \tag{4}$$

Eq. (4) takes values in the unit real range, where 0 indicates maximum similarity and 1 maximum dissimilarity; intermediate values provide a degree of similarity proportional to the cosine of the angle between the two vectors ξ_i and ξ_j.

By combining the weights w_{ij} and the functions defined for each pair of distinct indices $\{i, j\}$ in (4), the problem of finding a bipartition $\{\mathcal{A}, \mathcal{B}\}$ of $\mathcal{S}$, can be formulated as an optimization problem on the hypercube $D = [0, 2\pi]^L$ with the objective function to be maximized given by

$$\begin{aligned} J(\theta_1, \dots, \theta_L) &= \sum_{i<j} w_{ij}\, d_{i,j}(\xi_i, \xi_j) \\ &= \sum_{i<j} w_{ij} \frac{1 - \cos(\theta_i - \theta_j)}{2} \\ &= c - \frac{1}{2} \sum_{i<j} w_{ij} \cos(\theta_i - \theta_j), \end{aligned}$$

where $c = \frac{1}{2} \sum_{i<j} w_{ij}$.

In this equation, the term to be maximized appears (up to a constant) with the negative sign, which is equivalent to minimize the same functional with inverted sign (which with some abuse of notation we denote with the same symbol):

$$J(\theta_1, \dots, \theta_L) = \frac{1}{2} \sum_{i<j} w_{ij} \cos(\theta_i - \theta_j). \tag{5}$$

Intuitively, each term $w_{ij}\cos(\theta_i - \theta_j)$ in the previous sum contributes to the magnitude of J depending on the value of the weight w_{ij}. In fact, when $w_{ij} \approx 0$, its product with $\cos(\theta_i - \theta_j)$ allows for free choice of θ_i and θ_j; on the contrary when $w_{ij} > 0$, the amplitude of $\cos(\theta_i - \theta_j)$ is forced to be as close as possible to zero, and consequently to interpose an angle π between ξ_i and ξ_j, thus forcing their separation onto opposite semicircular loci of points.

The function in Eq (5) is in general non-convex, hence finding an exact global solution is an NP-hard problem. One way to deal with non-convexity is to relax the goal from finding the global minimum to looking for a local minimum using, for example, a local search technique based on a discrete-time dynamical system with asynchronous update rule and minimum-energy state estimation, as in the following formalization.

Let us associate to the function in Eq (5) a local updating rule $T = \left(T_k\right)_{k=1}^{L}$ on hypercube D, i.e. $T(\theta) = \left(T_1(\theta_1), \dots, T_L(\theta_L)\right)$ such that

$$T_k(\theta_k) = \underset{\theta_k \in [0,2\pi]}{\operatorname{argmin}} \left\{ C_k \cos\theta_k + S_k \sin\theta_k \right\}, \tag{6}$$

where

$$C_k = \sum_{k \neq j} w_{kj} \cos\theta_j \quad \text{and} \quad S_k = \sum_{k \neq j} w_{kj} \sin\theta_j. \tag{7}$$

For the system in Eq (6) the point $\theta_k^* \in [0, 2\pi]$ is an equilibrium point if and only if

$$T_k(\theta_k^*) = \theta_k^*, \quad k = 1, \dots, L,$$

implying that

$$T(\theta^*) = \theta^*, \quad \theta^* \in D. \tag{8}$$

To deal with (8), we introduce the discrete time nonlinear autonomous system with asynchronous dynamics (6) defined through the recurrent relation

$$\theta(t+1) = T(\theta(t)), \quad \theta(0) = \theta_0, \quad t \in \mathbb{N}, \tag{9}$$

where $\theta(t) \in D$ is the state at time t. In asynchronous updating, the components of the current state vector $\theta(t)$ are updated one at a time according to Eq. (6), so as to produce the new state vector $\theta(t+1)$. In order to show that system in Eq (9) converges to an asymptotically stable fixed point $\theta^* \in D$ that locally minimize the function J on D we do the following considerations.

The equilibrium point $\theta^* \in D$ is said to be (locally) asymptotically stable in the sense of Lyapunov if for every $\epsilon > 0$, there exists some $\delta > 0$ such that $\|\theta(0) - \theta^*\| < \delta$ implying $\|\theta(t) - \theta^*\| < \epsilon$ for all $t \in \mathbb{N}$, and then $\lim_{t \to +\infty} \theta(t) = \theta^*$ [15]. That is, if the state of the system is close to the equilibrium initially, it always stays close to the equilibrium. If, in addition, the state converges to the equilibrium, θ^*, it is said to be asymptotically stable in the sense of Lyapunov.

To prove this stability property for the system in Eq (9), the following lemma is needed.

Lemma 2.1. *Assume $W = \left(w_{ij}\right)$ is a $L \times L$ symmetric matrix, and that θ_j, $j = 1, \dots, L$, are fixed, then the difference*

$$\Delta_k J = J(\theta_1, \dots, \theta_k', \dots, \theta_L) - J(\theta_1 \dots, \theta_k, \dots \theta_L) \leq 0$$

if and only if

$$\begin{cases} \pi - \alpha \leq \theta_k' \leq 2\pi, & \text{if } \alpha \leq \pi \\ 0 \leq \theta_k' \leq 2\pi - \alpha, & \text{if } \alpha > \pi \end{cases}$$

where $\cos\alpha = C_k / \sqrt{C_k^2 + S_k^2}$, and C_k, S_k are defined in (7). Moreover:

$$\underset{\theta_k' \in [0,2\pi]}{\operatorname{argmax}} \Delta_k J = \begin{cases} \alpha + \pi, & \text{if } \alpha \leq \pi \\ \alpha - \pi, & \text{if } \alpha > \pi. \end{cases}$$

Proof. Due to the symmetry of W we start by rewriting the function J in (5) in the form:

$$\begin{aligned} J(\theta_1, \dots, \theta_L) &= \frac{1}{4} \sum_{i,j=1}^{L} w_{ij} \cos(\theta_i - \theta_j) \\ &= \frac{1}{4} \sum_{j=1}^{L} w_{kj} \cos(\theta_k - \theta_j) + c, \end{aligned}$$

where $c = \frac{1}{4} \sum_{i,j \neq k} w_{ij} \cos(\theta_i - \theta_j)$.

Using the variables C_k and S_k defined in (7), we can write the difference as

$$\begin{aligned} \Delta_k J &= J(\theta_1, \dots, \theta_k', \dots, \theta_L) - J(\theta_1 \dots, \theta_k, \dots \theta_L) \\ &= \frac{1}{4} \sum_j w_{kj} \left(\cos\theta_k' \cos\theta_j + \sin\theta_k' \sin\theta_j \right) \\ &\quad - \frac{1}{4} \sum_j w_{kj} \left(\cos\theta_k \cos\theta_j + \sin\theta_k \sin\theta_j \right) \\ &= \frac{1}{4} \left(C_k \cos\theta_k' + S_k \sin\theta_k' - C_k \cos\theta_k - S_k \sin\theta_k \right). \end{aligned}$$

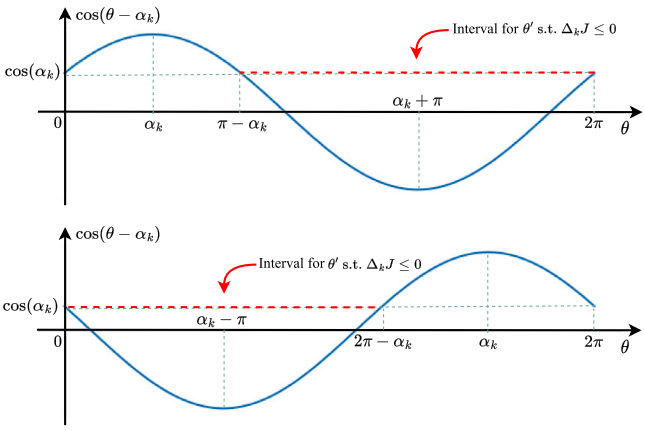


Fig. 9. The two cases of intervals for θ' where $\Delta_k J \leq 0$.

By multiplying and dividing $\Delta_k J$ by $Z_k = 4\sqrt{C_k^2 + S_k^2}$ and setting $\cos\alpha_k = C_k / Z_k$, $\sin\alpha_k = S_k / Z_k$, we have

$$\begin{aligned} \Delta_k J &= Z_k (\cos\alpha_k \cos\theta_k' + \sin\alpha_k \sin\theta_k' \\ &\qquad - \cos\alpha_k \cos\theta_k - \sin\alpha_k \sin\theta_k) \\ &= Z_k \left(\cos(\theta_k' - \alpha_k) - \cos(\theta_k - \alpha_k) \right). \end{aligned}$$

Therefore, assuming that $\alpha_k \in [0, 2\pi]$, since $Z_k > 0$, then $\Delta_k J \leq 0$ if and only if

$$\begin{cases} \pi - \alpha_k \leq \theta_k' \leq 2\pi, & \text{if } \alpha \leq \pi \\ 0 \leq \theta_k' \leq 2\pi - \alpha_k, & \text{if } \alpha > \pi. \end{cases}$$

This scenario is clearly shown in Fig. 9 where the two cases of intervals for θ_k' are highlighted with a dashed red line.

In particular, concerning the minimum of the difference $\Delta_k J$, the following holds:

$$\begin{aligned} \hat{\theta}_k &= \underset{\theta_k' \in [0,2\pi]}{\operatorname{argmin}} Z_k \left(\cos(\theta_k' - \alpha_k) - \cos(\theta_k - \alpha_k) \right) \\ &= - \underset{\theta_k' \in [0,2\pi]}{\operatorname{argmin}} Z_k \left(1 + \cos(\theta_k - \alpha_k) \right) \end{aligned}$$

i.e.,

$$\hat{\theta}_k = \begin{cases} \alpha_k + \pi, & \text{if } \alpha_k \leq \pi \\ \alpha_k - \pi, & \text{if } \alpha_k > \pi. \end{cases} \quad \square$$

In order to use the fundamental results of Lyapunov stability theory for the system (9), we observe that $J : \mathcal{D} \to \mathbb{R}$ is Lipschitz continuous on $\mathcal{D}$, since the same holds for the trigonometric function $\cos : [0, 2\pi] \to [-1, 1]$, being its Lipschitz constant 1. By Lemma 2.1 and the continuity of function (5), the main theorem on discrete-time nonlinear dynamical system states that J is a Lyapunov function for system (9) [15]. In fact, for $J(\theta) - c$ (see definition (5) for constant c) it holds:

$$\begin{aligned} J(0) &= 0, \\ J(\theta) &> 0, \quad \theta \in \mathcal{D} - \{0\}, \\ J(T(\theta)) - J(\theta) &\leq 0, \quad \theta \in \mathcal{D}. \end{aligned}$$

2.3.1. Circle clustering algorithm

To find the solution to system (9), we apply the recursive schema sketched in Algorithm 1. The procedure involves two stages. The first, iteratively computes an approximate solution to (9), generating a finite trajectory (or sequence) $\theta(0), \theta(1), \theta(2), \dots, \theta(\hat{t})$, starting at an arbitrary

point $\theta(0) = \theta_0$, until the condition $|J(T(\theta(\hat{t}))) - J(\theta(\hat{t}))| < \varepsilon$, for a fixed $\varepsilon > 0$, is met. Actually, the algorithm uses a more efficient stopping condition given by the difference between two iterations, i.e., $|\theta(k+1) - \theta(k)| < \varepsilon$. The second, given the approximate solution $\theta(\hat{t}) \approx \theta^*$ to (5), outputs two clusters $\{A, B\}$ by easily identifying the best separating hyperplane on the circle which splits the vectors $\xi_k = (\cos\theta_k, \sin\theta_k) \in \mathbb{S}_2$ in the best possible way.

Algorithm 1: CircleClustering

Data: The weight matrix W and a real $\varepsilon > 0$.
Result: Bipartition $\{A, B\}$ of S.
/* STAGE 1 */
$\forall k \quad \theta_k \leftarrow$ rand in $[0, 2\pi]$;
for $k := 1$ **to** L **do**
  $C_k \leftarrow \sum_j w_{kj} \cos\theta_j$;
  $S_k \leftarrow \sum_j w_{kj} \sin\theta_j$;
end
$cond \leftarrow True$;
while $cond$ **do**
  $cond \leftarrow False$;
  for $k := 1$ **to** L **do**
    $\hat{\theta}_k \leftarrow \operatorname{argmin}_{\gamma\in[0,2\pi]} \{C_k \cos\gamma + S_k \sin\gamma\}$;
    if $|\hat{\theta}_k - \theta_k| > \varepsilon$ **then**
      $cond \leftarrow True$;
      for $j := 1$ **to** L **do**
        $C_j \leftarrow C_j + w_{kj}(\cos\hat{\theta}_k - \cos\theta_k)$;
        $S_j \leftarrow S_j + w_{kj}(\sin\hat{\theta}_k - \sin\theta_k)$;
      end
      $\theta_k \leftarrow \hat{\theta}_k$;
    end
  end
end
/* STAGE 2 */
for $k := 1$ **to** L **do**
  $\xi_k^{*\perp} \leftarrow (-\sin\theta_k^*, \cos\theta_k^*)$;
  $g_k(\theta^*) \leftarrow \sum_{i<j} w_{ij} \operatorname{sign}\left(\sin(\theta_j^* - \theta_k^*)\sin(\theta_i^* - \theta_j^*)\right)$;
end
$\kappa \leftarrow \operatorname{argmin}_{k\in\mathcal{L}} g_k(\theta^*)$;
$A \leftarrow \{j \mid \sin(\theta_j^* - \theta_\kappa^*) \geq 0\}$;
$B \leftarrow \{j \mid \sin(\theta_j^* - \theta_\kappa^*) < 0\}$;

Specifically, observe that the local minimum θ^* of the objective function (5) achieved in STAGE 1 satisfies the following equilibrium conditions:

$$\frac{\partial f}{\partial \theta_k}\Big|_{\theta^*} = \sum_j w_{kj} \sin(\theta_j^* - \theta_k^*) = 0, \quad \forall k = 1, \ldots, L.$$

Taking the orthogonal vector $\xi_k^{*\perp} = (-\sin\theta_k^*, \cos\theta_k^*) \in \mathbb{S}_2$ of ξ_k^*, for each k we obtain the same expression as above:

$$\sum_j w_{kj}\, \xi_j^{*\perp} \cdot \xi_k^* = \sum_{i<j} w_{ij} \sin(\theta_j^* - \theta_k^*),$$

meaning that in order to preserve the local minimum θ^* in the split, the bipartition of indexes $j \in \mathcal{L}$ (to get the two clusters) can be done by seeking the one, say ξ_κ^*, which acts as the best separating hyperplane, that is,

$$\kappa = \operatorname*{argmin}_{k\in\mathcal{L}} \sum_{i<j} w_{ij} \operatorname{sign}\left[\sin(\theta_j^* - \theta_k^*)\sin(\theta_i^* - \theta_k^*)\right], \tag{10}$$

where $\operatorname{sign}[\cdot]$ represents the sign function. This process finally leads to the bipartition $\{\mathcal{A}, \mathcal{B}\}$ of the original set of PSDs $S = \{S_1 \ldots, S_L\}$, where $S_i \in \mathcal{A} \Leftrightarrow \xi_i \in A$ and $S_i \in \mathcal{B} \Leftrightarrow \xi_i \in B$. This procedure is replicated for each patch within each frame window.

Fig. 10 provides a concrete example of the overall process which divides the PSDs associated to the patches into two clusters. The blue curve (left image) is the average PSD curve of the first cluster and the red one (middle image) is the average PSD curve of the second cluster. The circle in right image, depicts the fixed point θ^* which, in turn, is partitioned into blue and red points according to the minimization of (10). Notably, Fig. 10 shows a typical scenario where the power of the main pulsatile component of the BVPs is concentrated on a single frequency as in the good cluster (blue line). Conversely, in the bad cluster (red line), the spectra of the extracted signals are spread over a wide frequency range, as typically shown by noisy patterns. This example clearly illustrates the ability of CircleClustering to separate the pulsatile component (when present) from other elements (eg. noise, head movements, ambient light variation etc.).

2.3.2. Determining the good cluster

Fig. 11 shows a Gaussian fitting of the average PSD curves obtained in the previous step. It should be evident that when the average PSD is concentrated on a single frequency, the main lobe that contains most of the power of the PSD can be well approximated by a Gaussian function. Therefore, the cluster whose average PSD best fits a Gaussian function is chosen as the good one and is hence employed to perform the final BPM estimation. Eventually, Eq. (3) is applied to the average PSD of the good cluster (instead of to the single PSDs S^l) to determine the final BPM estimate.

2.3.3. The quest for a novel clustering algorithm

A general comment is deserved at this point, to relate CircleClustering to other directional data clustering methods, such as spherical k-means [27] and Von Mises clustering [28]. The first is a simple extension of the classical k-means for sparse unit vectors, while the latter is a generative model consisting of a mixture of von Mises–Fisher distributions, tailored for directional data distributed on the surface of a unit hypersphere and based on EM for estimating the parameters of the mixture model. Naturally, PSDs can be considered in all respects as data distributed on unitary hyperspheres whose relevant features are the directions and not the general magnitude. So, it makes sense to compare the techniques mentioned here at least on the basis of the operating principle. In short, the main reason to introduce a novel technique is that, when considering experimental data provided by the challenging rPPG datasets, none of the classic clustering techniques was able to separate the real pulsatile component from the multiform noise sources affecting it. Conversely, empirical evidence suggests that Algorithm 1 naturally produces a "good cluster" and a "bad cluster", the first comprising PSDs tightly localized around a common mean pulse rate, while the other cluster comprises PSDs more reminiscent of noise.

Furthermore, as to the computational demands, it is worth noting that the EM algorithm employed by Von Mises clustering is computationally expensive and does not cope well with the real-time constraints requested by many rPPG applications.

2.3.4. Computational complexity

As for the worst-case time complexity of the CircleClustering algorithm, note that the first stage of algorithm 1 is the most demanding, and within it the repetition of the **while** loop represents the most time-consuming block. Therefore, the following proposition can be proved.

Proposition 2.2. *Given an instance of size n and a real $\varepsilon > 0$, the worst-case running time of the algorithm* CircleClustering *is $\mathcal{O}(n^2/\varepsilon)$.*

Proof. Fix a sufficiently small real $\varepsilon > 0$. For an arbitrary trajectory $\theta(0), \theta(1), \ldots, \theta(\tau), \ldots$ of states visited by system (9) with corresponding units $k_0, k_1 \ldots, k_\tau \in [1..n]$ asynchronously updated at each step, by Lemma 2.1 we have the strictly decreasing sequence:

$$|\Delta_{k_0} J| > |\Delta_{k_1} J| > \cdots > |\Delta_{k_\tau} J| > \varepsilon.$$

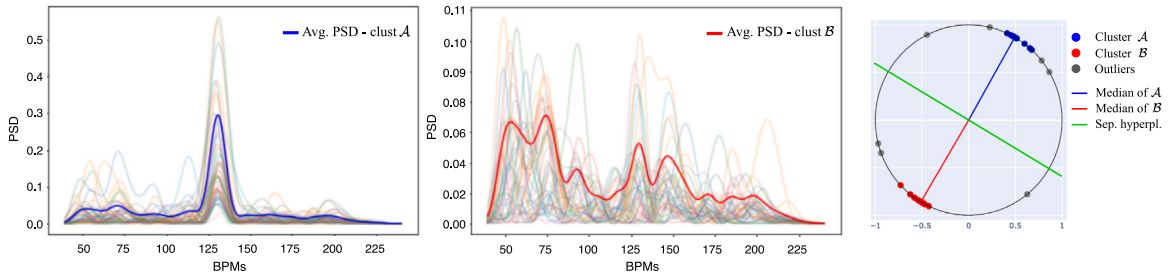


Fig. 10. Clusters $\{\mathcal{A}, \mathcal{B}\}$ (with average PSDs depicted in blue and red) achieved by Algorithm 1. The central circle represents the separation of elements on $\mathbb{S}_2$ yielded by CircleClustering.

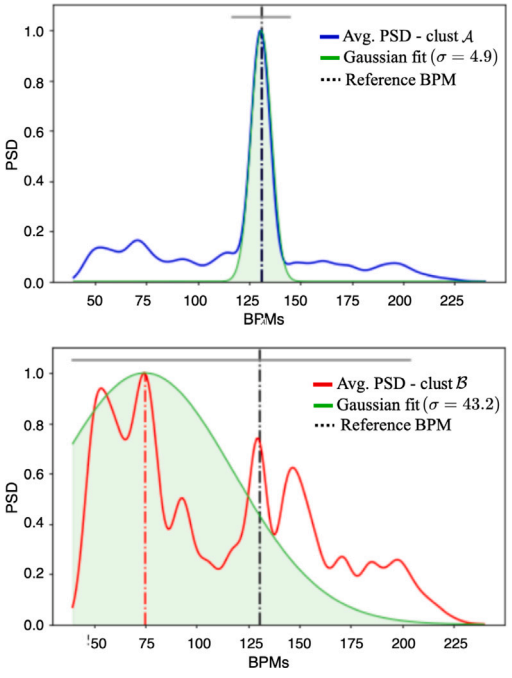


Fig. 11. Average PSD curves (blue and red) of the two clusters. The Gaussian fitting is shown by a green shadow curve.

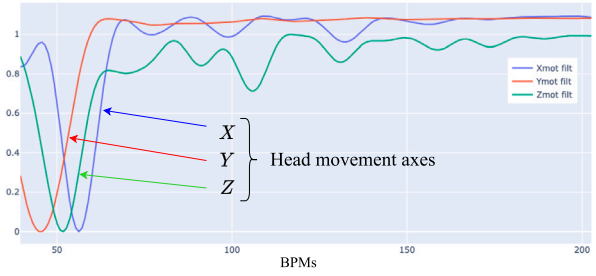


Fig. 12. Notch-shaped filters for head movements, one for each axes X (horizontal), Y (vertical) and Z (depth).

Consider that in each term of the previous sequence the factor $Z_k = 4\sqrt{C_k^2 + S_k^2}$ which can be expanded as

$$
\begin{aligned}
Z_k^2/4 &= \Big(\sum_{j=1}^{n} w_{kj}\cos\theta_j\Big)^2 + \Big(\sum_{j=1}^{n} w_{kj}\sin\theta_j\Big)^2 \\
&= \sum_{i,j} w_{ki}w_{kj}(\cos\theta_i\cos\theta_j + \sin\theta_i\sin\theta_j) \\
&= \sum_{i,j} w_{ki}w_{kj}\cos(\theta_i - \theta_j).
\end{aligned}
$$

By denoting with $\bar{w}$ the maximum entry of the distance matrix W, the previous expansion can be upper bounded by $\bar{w}^2n^2$, thus implying $Z_k \leq 2\bar{w}n$. Since in the worst case we have at least one update in each iteration of the **while** loop and its cost is proportional to n, we conclude that the worst-case running time is $\mathcal{O}(n^2/\varepsilon)$. □

In terms of actual complexity, it is challenging to provide absolute metrics due to the strong dependency on implementation and available computational resources. However, it can be noted that, using a standard PC with a modern CPU and no GPU requirement, many of the clustering methods used in the approach allow for real-time computations. Specifically, the most computationally demanding part is video analysis, such as extracting landmarks and tracking them across frames. Approximately 50% of the total processing time is allocated to this operation for HD video. In contrast, the clustering process, particularly when using the CHROM and POS methods (the fastest), takes up around 2% of the time, ensuring that time constraints are met in almost all test scenarios.

2.4. Motion analysis

It is clearly understood that the movement of the head substantially affects the geometric structure between the light source, the skin surface, and the camera [8]. It also represent a very insidious type of noise as its spectral components can overlap those of the heart beat, especially when the motion is regular and its spectral energy falls within the frequency band of interest. To deal with this drawback, we devise a family of filters empirically derived from the displacement of certain reference landmarks not subjected to deformation, such as those located on the nose or forehead. The rationale is that of equalization through cut or notch-shaped filters, which suitably reshape the PSDs collected in the set $\mathcal{S} = \{S^l(\omega) \in \mathbb{R}^T : l \in \mathcal{L}\}$ yielded by patches. This step is performed prior to the clustering procedure.

A typical cut filter pattern triad used to counteract movement is shown in Fig. 12, one for each axis of motion, namely X (horizontal), Y (vertical), and Z (depth). In particular, the PSDs of the movement, projected on the axes and denoted by $\mathcal{S}_X(\omega)$, $\mathcal{S}_Y(\omega)$ and $\mathcal{S}_Z(\omega)$ respectively, are calculated on the basis of the changes in the positions of the landmarks within the bounding box that crops the face.

This step is an essential premise for calculating the filter parameters, that are the amplitude and cutoff frequency, which will shape the frequency response of the filters themselves. In other words, the frequency response of each of the three filters is obtained as a complement to the normalized movement PSDs with unit gain on the maxima, thus giving rise to the model:

$$H_*(\omega) = 1 - \mathcal{S}_*(\omega)$$

where $_*$ represents each axes X, Y, or Z.

Under such assumptions and by activating the motion filters only when the energy of the motion indicated by the landmarks exceeds an empirically predetermined threshold, each PSD $S^l(\omega)$ at landmark level is analytically remodeled as follows:

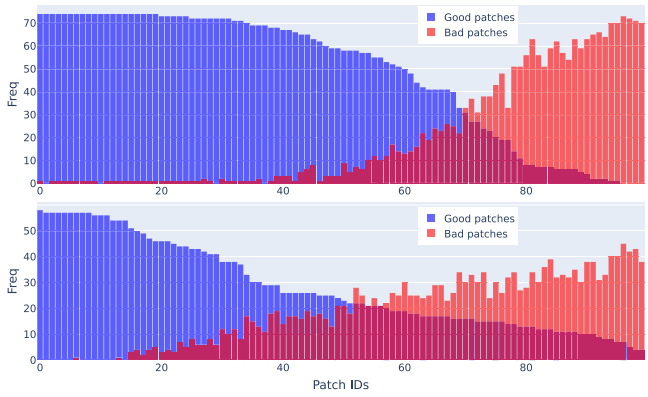


Fig. 13. Histograms of the distribution of good and bad patches for two video samples from two distinct datasets.

$$\hat{S}^l(\omega) = H_X(\omega)\, H_Y(\omega)\, H_Z(\omega)\, S^l(\omega).$$

The clustering procedure previously described is eventually applied to the set $\mathcal{S} = \{\hat{S}^l(\omega) \in \mathbb{R}^T : l \in \mathcal{L}\}$.

2.5. Parameter setting and workflow

Patch detection and tracking is carried out via the pyVHR framework [9,10], which relies on the MediaPipe model [29] for the extraction of facial landmarks. This model identifies and tracks up to 468 landmarks on the face, which are used to exclude non-skin regions, such as the eyes and mouth, and to define the set of patches. In all our experiments, we employ a set of 100 patches uniformly sampled across the subject's face.

All subsequent processing, including filtering, actual rPPG computation and spectral estimation are carried out via the pyVHR framework. The code to reproduce the method and its results is available on github.[2]

To clarify the workflow and key parameters involved in the bipartition process executed by CircleClustering for final BPM inference using Gaussian analysis, we summarize the approach as follows. At each time t, after generating the two clusters $\mathcal{A}_t$ and $\mathcal{B}_t$ (as shown in Fig. 10), a decision process is initiated. This decision is primarily based on the normality of the sum of power spectra, which have been divided into the two clusters C_t^G (good cluster) and C_t^B (bad cluster). The Gaussian fitting provides the mean μ_g^G and std σ_g^G for the good cluster, as well as μ_t^B and σ_t^B for the bad cluster.

Although this process concludes by providing a BPM value at each time t equal to μ_t^G, it is insightful to analyze the informational contribution that individual patches, particularly their aggregation, dynamically bring to the process. Two elements are of importance in this simple empirical analysis we present: the frequency of patches in the two clusters and the Gaussian fitting parameters in the successive time windows.

Examples of frequency distributions are provided in Fig. 13. Two distinct videos, sourced from different datasets, were analyzed, and the corresponding patch frequencies are reported. Each patch contributes to the overall video by providing salient information, either good or bad. Notably, a given patch may not consistently fall into the good or bad cluster across frames in the same video, and its classification can differ when analyzed in a different video. These distributions further illustrate the varying influence that individual patches have on the final analysis.

[2] https://github.com/phuselab/pyVHR.

Fig. 14. Saliency maps based on patch importance for two sample from two distinct datasets.

Notably, this allows to establish a quantitative measure that scores the importance of a patch belonging to a cluster. To this end, we evaluate how likely is the Ground-Truth value g_t under the obtained Gaussian fitting of both the good and bad clusters:

$$p(g_t \mid \mu_t^*, \sigma_t^{*2}) = \mathcal{N}(g_t \mid \mu_t^*, \sigma_t^{*2}), \qquad \forall t = 1, \ldots, T,$$

where $*\in \{G, B\}$, and $\mathcal{N}$ represents the Gaussian distribution of mean μ_t^* and variance σ_t^{*2}. The rationale behind this is that typically $p(g_t \mid \mu_t^G, (\sigma_t^G))^2 > p(g_t \mid \mu_t^B, (\sigma_t^B)^2)$.

In Eq. (11) we combine the above quantity with the frequency with which a patch is included in a cluster. The combination of patch frequencies and Gaussian likelihoods over time, here referred to as temporal saliency, yields the overall saliency measure S_j^* for each patch j:

$$S_j^* = \sum_{t=1}^{T} \underbrace{p(g_t \mid \mu_t^*, \sigma_t^{*2})}_{\text{temporal saliency}} \cdot \underbrace{\mathbb{1}_{C_t}^*(j)}_{\text{patch freq}} \tag{11}$$

where $\mathbb{1}_{C_t}^*$ is the indicator function of the subset $C_t^* \subseteq S_t^*$ and T is the total number of time steps.

Fig. 14 illustrates the saliency S^* for two cases taken from two different datasets, corresponding to the frequencies shown in Fig. 13. The images highlight the face regions typically recognized as belonging to either the good or bad cluster.

3. Experimental work

This section presents the experimental setup and empirical analysis on multiple benchmark corpora. After briefly introducing the adopted datasets, we perform extensive experimental analyses to demonstrate the consistent improvement yielded by the proposed approach. To this end, we consider five rPPG recovery methods and apply CircleClustering to each of them.

3.1. Datasets

The datasets used for comparison are chosen to give a complete picture of the abilities of the proposed technique to handle a wide variety of challenging situations for the task of remote heart rate recovery. Specifically, four datasets have been selected due to their wide availability and use in the rPPG-related literature. Moreover these corpora allow to explicitly test the robustness of our approach to various lighting conditions (ambient light, halogen lamp, etc.), movement patterns (head movements, camera movements, movements induced by cardiac activity, talking heads, etc.) and overall contexts (laboratory, gym, in the wild, etc.). These are briefly described below:

PURE [30]. This database comprises 10 subjects (8 male, 2 female) that were recorded in 6 different setups, resulting in a total number of 60 sequences of 1 min each. Lighting condition was frontal daylight, with clouds changing illumination conditions slightly over time. Six different setups have been recorded: Steady (S); Talking (T); Slow translation (ST); Fast translation (FT); Small rotation (SR); Medium rotation (MR).

Table 1
Average performance comparisons on all the datasets using MAE and quality metrics (PCC and MAX) for holistic and patched approaches collecting results from core methods. The best values for each metric is shown in **bold**.

Dataset	PatchClustering			PatchMedian			Holistic		
	MAE ↓	PCC ↑	MAX ↓	MAE ↓	PCC ↑	MAX ↓	MAE ↓	PCC ↑	MAX ↓
PURE Steady	**0.92**	**0.91**	**3.07**	1.27	0.80	5.15	15.37	0.35	72.88
PURE Talking	**2.85**	**0.72**	**17.58**	4.09	0.70	18.23	5.57	0.56	34.37
PURE Small Translation	**0.72**	**0.95**	**2.94**	0.93	0.90	3.63	8.74	0.47	55.01
PURE Fast Translation	**0.86**	**0.92**	**2.91**	1.19	0.86	4.70	5.37	0.52	31.06
PURE Small Rotation	**1.92**	**0.79**	**5.11**	2.24	0.75	9.57	5.86	0.44	35.96
PURE Medium Rotation	**3.02**	**0.76**	**6.72**	3.69	0.70	12.43	3.78	0.49	26.25
UBFC1	**0.75**	**0.95**	**5.82**	1.71	0.81	15.16	9.97	0.47	54.67
UBFC2	**3.61**	**0.84**	**14.64**	5.27	0.70	24.94	10.55	0.47	48.74
LGI-PPGI Resting	**1.97**	0.53	**8.27**	2.47	**0.60**	10.58	11.65	0.32	61.51
LGI-PPGI Rotation	**4.49**	0.26	**22.33**	4.12	**0.38**	23.66	8.46	0.18	66.16
LGI-PPGI Talk	12.98	**0.10**	**37.67**	**11.35**	0.01	41.18	11.41	0.02	54.34
LGI-PPGI Gym	**14.42**	**0.75**	**45.68**	19.68	0.46	70.64	18.10	0.45	80.33
ECG-FITNESS Rowing Ambient	**7.55**	**0.36**	**22.10**	13.33	0.17	37.73	10.0	0.33	44.21
ECG-FITNESS Speaking Ambient	**18.25**	**0.43**	**34.60**	44.31	0.08	65.55	36.74	0.09	75.06
ECG-FITNESS Rowing Halogen	15.62	**0.33**	**31.76**	21.53	0.12	45.25	**15.32**	0.23	53.59
ECG-FITNESS Speaking Halogen	**33.20**	**0.26**	**52.79**	55.44	0.12	81.32	46.25	0.09	86.45
ECG-FITNESS Elliptical Ambient	**23.33**	**0.40**	**39.26**	35.17	0.30	57.85	25.29	0.26	66.86
ECG-FITNESS Bike Ambient	**28.30**	**0.45**	**49.91**	38.67	0.15	62.64	29.60	0.22	68.85
Global Mean	**9.71**	**0.59**	**22.57**	14.80	0.48	32.79	15.41	0.33	56.52

UBFC [31]. This dataset is composed of 50 videos divided into two subsets: the first one, UBFC1 is composed by 8 videos, in which participants were asked to sit still; the second one, UBFC2 is composed by 42 videos, in which participants were asked to play a time sensitive mathematical game that aimed at augmenting their heart rate while simultaneously emulating a normal human–computer interaction scenario.

LGI-PPGI [32]. This database is designed to estimate the heart rate from uncompressed face videos acquired in the wild. It is recorded in four different sessions: (1) a resting scenario with neither head motion or illumination changes, (2) head movements are allowed (with static lighting), (3) a more ecological setup, where people are recorded while performing exercises on a bicycle ergometer in a gym; (4) urban conversations are recorded including head and camera motions as well as natural varying illumination conditions (in the wild setup).

ECG-fitness [33]. Its collects a realistic corpus of subjects performing physical activities on fitness machines: 17 subjects (14 male, 3 female) performing 4 different activities (speaking, rowing, exercising on a stationary bike, and on an elliptical trainer). Three lighting setups were used, natural light coming from a nearby window, 400 W halogen light and 30 W led light. The dataset covers the following challenges: large subject's motion (possibly periodic) on all three axis, rapid motions inducing motion blur, strong facial expressions, wearing glasses,non-uniform lighting, light interference, atypical non-frontal camera angles.

3.2. rPPG methods

We select the following signal processing based rPPG methods for pulse extraction:

ICA [34]. Decomposition based on blind source separation (BSS) to achieve independent components from temporal RGB mixtures.

PCA [35]. Statistical technique for extracting a subset of uncorrelated components from temporal RGB traces.

CHROM [24]. Chrominance-based method to perform color channel normalization to overcome distortions.

POS [8]. It leverages on a plane orthogonal to the skin-tone in the temporally normalized RGB space.

LGI [32]. It provides features invariant to action and motion based on differentiable local transformations.

3.3. Experiments and results

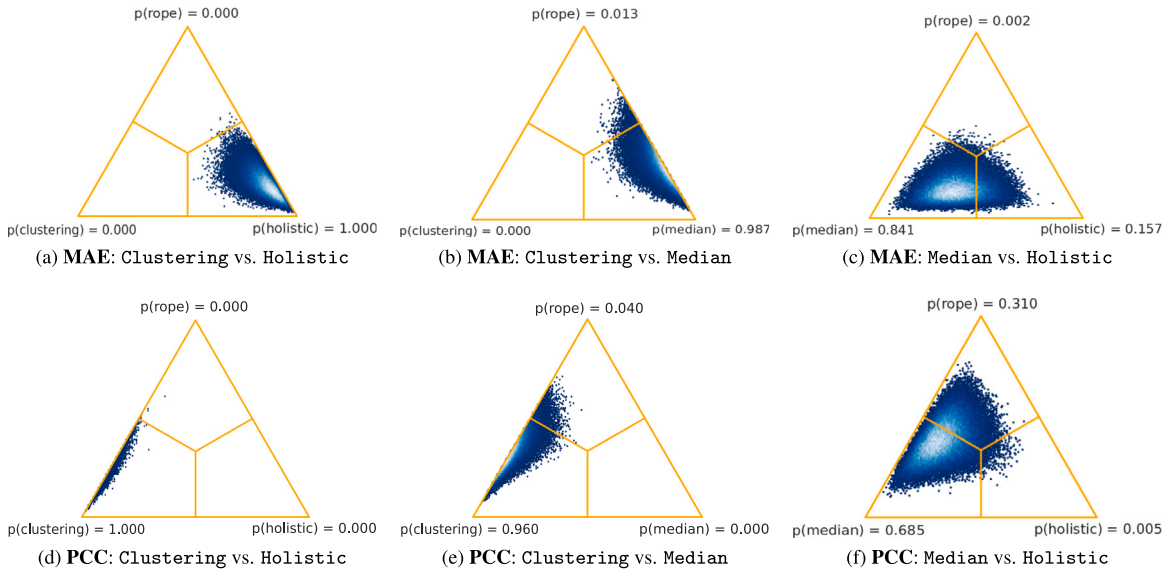
For each video clip, we extract 100 facial landmarks with their corresponding RGB traces. Subsequently, these data are segmented into 8-seconds windows, and rPPG signals are estimated accordingly. A bandpass filter is applied to each rPPG estimate with a bandwidth of 40–240 BPM (which corresponds to 0.65–4 Hz). CircleClustering is then applied to each window, and a BPM estimate is provided as reported in Section 2.2. We refer to this procedure as the `PatchClustering` method. For comparison purposes, the median BPM across all patches (`PatchMedian`) and the prediction from the holistic approach (`Holistic`) are computed. To ensure a fair comparison, all parameters were kept consistent during the processing of videos belonging to all treated datasets.

As for the quantitative assessment, the following metrics were computed: the mean absolute difference (MAE) between the main pulsatile component of the estimated rPPG signal and the PPG ground truth, the Pearson's correlation coefficient (PCC), and the maximum error difference (MAX). While PCC is essentially a normalized measure of the covariance between two quantities, MAX captures the magnitude of the maximum outlier present in the provided estimates.

Table 1 shows at a glance the comparisons between the computational approaches discussed above, i.e. `PatchClustering`, `PatchMedian` and `Holistic` in terms of the adopted metrics. Each metric has been computed for every method across each dataset; in order to better investigate the robustness of the method across various conditions, each dataset is split following the provided experimental trials. The obtained values are then averaged. This allows us to highlight the average impact of the `PatchClustering` algorithm in comparison to the two commonly adopted approaches of estimating heart rate from rPPG signals.

As a general trend, all experiments seem to demonstrate a superior performance of the `PatchClustering` approach compared to the `PatchMedian` or `Holistic` approaches. Specifically, the MAE error increases by 52.4% for `PatchMedian` with respect to `PatchClustering` and by 58.7% for the `Holistic` approach, as can be seen by inspecting the global mean (cfr. Table 1, last row).

We substantiate this crude summary employing a proper statistical assessment technique. More precisely, we verify whether the differences in terms of a given metric are statistically significant or are drawn by chance. Typically, this involves the adoption of Null Hypothesis Statistical Testing (NHST) procedures, often used for rigorous performance evaluation of classification algorithms [36]. Recently, [37] proposed the adoption of Bayesian estimation techniques to assess



(a) **MAE**: `Clustering` vs. `Holistic` (b) **MAE**: `Clustering` vs. `Median` (c) **MAE**: `Median` vs. `Holistic`

(d) **PCC**: `Clustering` vs. `Holistic` (e) **PCC**: `Clustering` vs. `Median` (f) **PCC**: `Median` vs. `Holistic`

Fig. 15. Posterior samples for the Bayesian Sign-Rank Test on the simplex for the MAE (top row) and PCC (bottom row) metrics. Each plot shows the probability for an approach to yield higher values of the metric if compared with another on multiple datasets.

the significance of performance differences between machine learning algorithms on different datasets.

In the vein of [37], we gauge the eventual improvement of the proposed methods through a Bayesian non-parametric approach that directly extends the Wilcoxon signed-rank test [38]. Differently from frequentist NHST, Bayesian estimation allows to inspect the actual probability of a method to yield results that are either greater or equivalent to another. The equivalence is established via the so called Region of Practical Equivalence (ROPE) i.e. an interval inside which the differences of performance (for a given performance metric) between two approaches can be considered of negligible magnitude. [39] suggests that ROPE should be set as half the value of what can be considered as a negligible magnitude according to the metric at hand. Consequently for the MAE, we consider the differences below 1 BPM as negligible, hence we set $rope = 0.5$. On the other hand, for the PCC metric, we set $rope = 0.05$ (half of a negligible correlation, as suggested by [40]). The results are depicted in Fig. 15, which reports the posterior distribution samples of the Bayesian Sign-Rank Test on the simplex for the MAE (top row) and PCC (bottom row) metrics. Each vertex of the triangle represents the case where an approach is either more probable to produce higher values of the specified metric w.r.t. the other or equivalent.

Results can be summarized as follows: when comparing the `PatchClustering` method with the `Holistic` one on multiple datasets, the `Holistic` method yields higher MAE values with probability 1 (Fig. 15(a)); the `PatchMedian` approach gives higher MAE values than `PatchClustering` with probability 0.987 (Fig. 15(b)); `PatchMedian` returns higher MAE values than `Holistic` with probability 0.841 (Fig. 15(c)). In summary, the analysis allows us to conclude that the proposed `PatchClustering` method significantly outmatches the `Holistic` and `PatchMedian` approaches. A similar conclusion can be drawn by inspecting the results on the PCC metric (Fig. 15(d–f)) with `PatchClustering` method yielding significantly higher PCC values than the others.

3.3.1. Comparison with deep learning-based approaches

The experimental results presented above show the added value introduced by `PatchClustering` as a systemic approach useful to enhance common signal processing-based methods for rPPG estimation. However, the general consensus in recent dedicated literature, proposes supervised DL-based approaches for rPPG estimation as the most accurate techniques. Consequently, in this section, a comparison between `PatchClustering` and five state-of-the-art neural rPPG approaches is provided. To this end, we refer to the results recently delivered by [41], which presents `rPPG-Toolbox`, a Python toolbox that implements and benchmarks different neural rPPG methods on a variety of different datasets.

Table 2
Average MAE values on the UBFC and PURE datasets obtained using CircleClustering with CHROM or POS as base methods, along with results from 5 state-of-the-art neural supervised models implemented in `rPPG-Toolbox`. In order to establish a fair comparison and ensure consistent evaluation across models, for neural approaches cross-dataset results are reported. The best values are highlighted in **bold**, and the second-best values are underlined.

Method	Train set	PURE	UBFC
`PatchClustering` (CHROM)	–	**0.92**	1.11
`PatchClustering` (POS)	–	1.85	1.41
TS-CAN [42]	PURE	–	1.29
	UBFC	3.69	–
PhysNet [43]	PURE	–	**0.98**
	UBFC	8.06	–
Physformer [12]	PURE	–	1.44
	UBFC	12.92	–
DeepPhys [11]	PURE	–	1.21
	UBFC	5.54	–
EfficientPhys [44]	PURE	–	2.07
	UBFC	5.47	–

Table 2 reports the results delivered by `PatchClustering` with two base rPPG methods, namely CHROM and POS, and the most representative supervised approaches implemented in `rPPG-Toolbox`. We present the MAE results obtained on two widely adopted datasets: PURE and UBFC. Both datasets are adopted in the benchmarks of both `rPPG-Toolbox` and `pyVHR`. For neural supervised models, we report cross-dataset results; this allows to evaluate the generalization abilities of the considered approaches, while ensuring a consistent comparison, as all adopted test-sets are coherent across all the considered benchmark models of Table 2.

The obtained results show an overall superiority of the unsupervised `PatchClustering` approach when compared to different state-of-the-art supervised neural methods. Specifically, employing `PatchClustering` with POS or CHROM as base rPPG estimation methods, consistently delivers better results on the PURE dataset and comparable results on UBFC. By resorting to averages across the two datasets,

`PatchClustering` achieves an average MAE of 1.38 BPM as opposed to the 7.14 BPM from neural approaches.

A more in-depth and systematic evaluation is reported in the "Appendix: Numerical Results (supplementary material)" where an extensive comparison of the base methods and two further supervised neural methods (HR-CNN and MTTS-CAN), is reported.

4. Conclusion

In this paper, we presented a novel PSD clustering algorithm, CircleClustering, which significantly improves the performance of traditional methods for camera-based heart rate measurement. The proposed technique integrates patch sampling with a non-parametric clustering approach on rPPG spectral data to estimate heart rates with high precision. CircleClustering leverages self-similarities and structural group information from skin patches tracked across video frames, clustering the spectral content of the pulse signals using a group Gaussian fitting applied to a bipartite set of PSDs. This is evident in two key aspects: the likelihood that the contribution of a patch matches the ground truth, and how often that patch is classified within the good cluster. These combined factors highlight the method's precision in isolating reliable pulse signals from noise, improving the accuracy of heart rate estimation.

The effectiveness of the proposed approach has been quantitatively evaluated through extensive experiments and robust statistical tests on a wide range of challenging datasets. Additionally, comparisons with various state-of-the-art supervised Deep Learning-based methods confirmed the overall superiority of the proposed technique in accurately recovering the ground truth pulsatile component. These results underscore the method's robustness and its potential advantages over traditional supervised approaches in real-world applications.

Future work will focus on extending the proposed approach by incorporating learning-based solutions. Specifically, instead of using a fixed similarity measure (such as cosine similarity) to run the CircleClustering algorithm, weakly supervised (deep) metric learning techniques [45] could be introduced. This adaptation could further enhance both the robustness and generalization capabilities of the approach. More on the application side, the proposed procedure may also be employed for more effective rPPG estimation from compressed videos, notoriously exhibiting faint physiological information [13] due to the disruption of the color variations associated to blood volume pulse variations. In such cases, the adoption of the proposed adaptive patch selection strategy may lead to a better choice of those areas of the subject's skin holding the most salient physiological information.

CRediT authorship contribution statement

Giuseppe Boccignone: Methodology, Investigation, Conceptualization. **Donatello Conte:** Visualization, Validation, Software, Resources, Methodology, Data curation. **Vittorio Cuculo:** Writing – original draft, Validation, Software, Funding acquisition, Data curation. **Alessandro D'Amelio:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology. **Giuliano Grossi:** Writing – review & editing, Supervision, Project administration, Formal analysis, Conceptualization. **Raffaella Lanzarotti:** Writing – original draft, Resources, Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and code are available on the site github.com/phuselab/pyVHR.

References

[1] D. McDuff, Camera measurement of physiological vital signs, ACM Comput. Surv. 55 (9) (2023) 1–40.

[2] L. Xu, J. Lien, H. Li, N. Gillian, R. Nongpiur, J. Li, Q. Zhang, J. Cui, D. Jorgensen, A. Bernstein, et al., Soli-enabled noncontact heart rate detection for sleep and meditation tracking, Sci. Rep. 13 (1) (2023) 18008.

[3] S. Zaunseder, A. Trumpp, D. Wedekind, H. Malberg, Cardiovascular assessment by imaging photoplethysmography–a review, Biomed. Eng./Biomed. Tech. 63 (5) (2018) 617–634.

[4] W. Mellouk, W. Handouzi, Cnn-lstm for automatic emotion recognition using contactless photoplythesmographic signals, Biomed. Signal Process. Control 85 (2023) 104907.

[5] G. Boccignone, A. D'Amelio, O. Ghezzi, G. Grossi, R. Lanzarotti, An evaluation of non-contact photoplethysmography-based methods for remote respiratory rate estimation, Sensors 23 (7) (2023) 3387.

[6] G. Boccignone, S. Bursic, V. Cuculo, A. D'Amelio, G. Grossi, R. Lanzarotti, S. Patania, Deepfakes have no heart: A simple rppg-based method to reveal fake videos, in: International Conference on Image Analysis and Processing, Springer, 2022, pp. 186–195.

[7] J. Wu, Y. Zhu, X. Jiang, Y. Liu, J. Lin, Local attention and long-distance interaction of rppg for deepfake detection, Vis. Comput. (2023) 1–12.

[8] W. Wang, A.C. den Brinker, S. Stuijk, G. de Haan, Algorithmic principles of remote ppg, IEEE Trans. Biomed. Eng. 64 (7) (2016) 1479–1491.

[9] G. Boccignone, D. Conte, V. Cuculo, A. D'Amelio, G. Grossi, R. Lanzarotti, An open framework for remote-ppg methods and their assessment, IEEE Access 8 (2020) 216083–216103, http://dx.doi.org/10.1109/ACCESS.2020.3040936.

[10] G. Boccignone, D. Conte, V. Cuculo, A. D'Amelio, G. Grossi, R. Lanzarotti, E. Mortara, Pyvhr: a python framework for remote photoplethysmography, PeerJ Comput. Sci. 8 (2022) e929, http://dx.doi.org/10.7717/peerj-cs.929.

[11] W. Chen, D. McDuff, Deepphys: Video-based physiological measurement using convolutional attention networks, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 349–365.

[12] Z. Yu, Y. Shen, J. Shi, H. Zhao, P.H. Torr, G. Zhao, Physformer: Facial video-based physiological measurement with temporal difference transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4186–4196.

[13] D.J. McDuff, E.B. Blackford, J.R. Estepp, The impact of video compression on remote cardiac pulse measurement using imaging photoplethysmography, in: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), IEEE, 2017, pp. 63–70.

[14] M. Zontak, M. Irani, Internal Statistics of a Single Natural Image, vol. 2011, CVPR, 2011, pp. 977–984.

[15] W.M. Haddad, V. Chellaboina, Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach, Princeton Un. Press, 2008.

[16] D.-Y. Kim, K. Lee, C.-B. Sohn, Assessment of roi selection for facial video-based rppg, Sensors 21 (23) (2021).

[17] P. Li, Y. Benezeth, K. Nakamura, R. Gomez, F. Yang, Model-based region of interest segmentation for remote photoplethysmography, in: 14th Int. Conf. on Comp. Vision Theory and Applications, SCITEPRESS, 2019.

[18] L.-M. Po, L. Feng, Y. Li, X. Xu, T.C.-H. Cheung, K.-W. Cheung, Block-based adaptive roi for remote photoplethysmography, Multimedia Tools Appl. 77 (6) (2018) 6503–6529.

[19] C.-H. Cheng, K.-L. Wong, J.-W. Chin, T.-T. Chan, R.H. So, Deep learning methods for remote heart rate measurement: A review and future research agenda, Sensors 21 (18) (2021) 6296.

[20] H. Xiao, T. Liu, Y. Sun, Y. Li, S. Zhao, A. Avolio, Remote photoplethysmography for heart rate measurement: A review, Biomed. Signal Process. Control 88 (2024) 105608.

[21] A. Ni, A. Azarang, N. Kehtarnavaz, A review of deep learning-based contactless heart rate measurement methods, Sensors 21 (11) (2021) 3719.

[22] B. Huang, C.-L. Lin, W. Chen, C.-F. Juang, X. Wu, A novel one-stage framework for visual pulse rate estimation using deep neural networks, Biomed. Signal Process. Control 66 (2021) 102387.

[23] Y.-Y. Tsou, Y.-A. Lee, C.-T. Hsu, S.-H. Chang, Siamese-rppg network: Remote photoplethysmography signal estimation from face videos, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 2066–2073.

[24] G. De Haan, V. Jeanne, Robust pulse rate from chrominance-based rppg, IEEE Trans. Biomed. Eng. 60 (10) (2013) 2878–2886.

[25] X. Li, J. Chen, G. Zhao, M. Pietikäinen, Remote heart rate measurement from face videos under realistic situations, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4264–4271, http://dx.doi.org/10.1109/CVPR.2014.543.

[26] M. Tarvainen, P. Ranta-aho, P. Karjalainen, An advanced detrending method with application to hrv analysis, IEEE Trans. Biomed. Eng. 49 (2) (2002) 172–175, http://dx.doi.org/10.1109/10.979357.

[27] I.S. Dhillon, D.S. Modha, Concept decompositions for large sparse text data using clustering, Mach. Learn. 42 (1–2) (2001) 143–175.

[28] A. Banerjee, I.S. Dhillon, J. Ghosh, S. Sra, Clustering on the unit hypersphere using von mises-fisher distributions 6 (2005) 1345–1382.

[29] Y. Kartynnik, A. Ablavatski, I. Grishchenko, M. Grundmann, Real-time facial surface geometry from monocular video on mobile gpus, 2019, arXiv preprint arXiv:1907.06724.
[30] R. Stricker, S. Müller, H.-M. Gross, Non-contact video-based pulse rate measurement on a mobile service robot, in: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, IEEE, 2014, pp. 1056–1062.
[31] S. Bobbia, R. Macwan, Y. Benezeth, A. Mansouri, J. Dubois, Unsupervised skin tissue segmentation for remote photoplethysmography, Pattern Recognit. Lett. 124 (2019) 82–90.
[32] C.S. Pilz, S. Zaunseder, J. Krajewski, V. Blazek, Local group invariance for heart rate estimation from face videos in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1254–1262.
[33] R. Spetlik, J. Cech, V. Franc, J. Matas, Visual heart rate estimation with convolutional neural network, in: British Machine Vision Conference, 2018.
[34] M.-Z. Poh, D.J. McDuff, R.W. Picard, Non-contact, Automated cardiac pulse measurements using video imaging and blind source separation., Opt. Exp. 18 (10) (2010) 10762–10774.
[35] M. Lewandowska, J. Rumiński, T. Kocejko, J. Nowak, Measuring pulse rate with a webcam - a non-contact method for evaluating cardiac activity, in: 2011 Federated Conference on Computer Science and Information Systems, FedCSIS, 2011, pp. 405–410.
[36] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[37] A. Benavoli, G. Corani, J. Demšar, M. Zaffalon, Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis, J. Mach. Learn. Res. 18 (1) (2017) 2653–2688.
[38] A. Benavoli, G. Corani, F. Mangili, M. Zaffalon, F. Ruggeri, A bayesian wilcoxon signed-rank test based on the dirichlet process, in: International Conference on Machine Learning, PMLR, 2014, pp. 1026–1034.
[39] J.K. Kruschke, T.M. Liddell, The bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective, Psychon. Bull. Rev. 25 (1) (2018) 178–206.
[40] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, Routledge, 2013.
[41] X. Liu, G. Narayanswamy, A. Paruchuri, X. Zhang, J. Tang, Y. Zhang, S. Sengupta, S. Patel, Y. Wang, D. McDuff, Rppg-toolbox: Deep remote ppg toolbox, 2023, arXiv:2210.00716.
[42] X. Liu, J. Fromm, S. Patel, D. McDuff, Multi-task temporal shift attention networks for on-device contactless vitals measurement, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 19400–19411.
[43] Z. Yu, W. Peng, X. Li, X. Hong, G. Zhao, Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 151–160.
[44] X. Liu, B. Hill, Z. Jiang, S. Patel, D. McDuff, Efficientphys: Enabling simple, fast and accurate camera-based cardiac measurement, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 5008–5017.
[45] B. Ghojogh, A. Ghodsi, F. Karray, M. Crowley, Spectral, probabilistic, and deep metric learning: Tutorial and survey, 2022, arXiv preprint arXiv:2201.09267.