Computer Methods and Programs in Biomedicine 240 (2023) 107712

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Event-based sampled ECG morphology reconstruction through self-similarity

Silvio Zanoli[a,1,*], Giovanni Ansaloni[a,2], Tomás Teijeiro[b,3], David Atienza[a,4]

[a] Embedded Systems Laboratory (ESL), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015
[b] Department of Mathematics, University of the Basque Country (UPV/EHU), Bilbao, Spain

ARTICLE INFO

Article history:
Received 6 July 2022
Revised 19 June 2023
Accepted 6 July 2023

Keywords:
Non-uniform sampling
Biosignal monitoring
Event-based
ECG
Morphology reconstruction
Dynamic time warping
ECG morphology

ABSTRACT

Background and Objective: Event-based analog-to-digital converters allow for sparse bio-signal acquisition, enabling local sub-Nyquist sampling frequency. However, aggressive event selection can cause the loss of important bio-markers, not recoverable with standard interpolation techniques. In this work, we leverage the self-similarity of the electrocardiogram (ECG) signal to recover missing features in event-based sampled ECG signals, dynamically selecting patient-representative templates together with a novel dynamic time warping algorithm to infer the morphology of event-based sampled heartbeats.

Methods: We acquire a set of uniformly sampled heartbeats and use a graph-based clustering algorithm to define representative templates for the patient. Then, for each event-based sampled heartbeat, we select the morphologically nearest template, and we then reconstruct the heartbeat with piece-wise linear deformations of the selected template, according to a novel dynamic time warping algorithm that matches events to template segments.

Results: Synthetic tests on a standard normal sinus rhythm dataset, composed of approximately 1.8 million normal heartbeats, show a big leap in performance with respect to standard resampling techniques. In particular (when compared to classic linear resampling), we show an improvement in P-wave detection of up to 10 times, an improvement in T-wave detection of up to three times, and a 30% improvement in the dynamic time warping morphological distance.

Conclusion: In this work, we have developed an event-based processing pipeline that leverages signal self-similarity to reconstruct event-based sampled ECG signals. Synthetic tests show clear advantages over classical resampling techniques.

1. Introduction

Nowadays, heightened life expectancy and unhealthy lifestyles make chronic diseases, and in particular chronic heart diseases, the leading cause of death worldwide [1]. Such conditions are long-lasting and not extensively observable inside hospitals, both for the short time of observation and the restricted set of activities a patient can do while hospitalized. Moreover, the need for a long observation period requires non-invasive solutions that impact the patient life as little as possible. This requirements make wearable solutions key for chronic disease monitoring.

One of the primary concerns of wearables is energy efficiency, as their main requirement is to function for the longest possible time while being unobtrusive to the patient. The energy budget of any battery operated device can be divided into four main categories: computation, storage, communication, and data acquisition. While computation, communications, and storage have been greatly studied and optimized in recent years [2–5], data acquisition remains a field scarcely explored. Nonetheless, the energy budget in modern wearable systems is highly affected by the signal acquisition component [6].

In their seminal work on sampling, Nyquist and Shannon defined an upper bound to the sampling rate, called Nyquist frequency [7], which is two times the maximum spectral component of the signal under analysis. Signals acquired following the

* Corresponding author.
E-mail address: silvio.zanoli@epfl.ch (S. Zanoli).
[1] orcid: 0000-0002-0316-1657
[2] orcid: 0000-0002-8940-3775
[3] orcid: 0000-0002-2175-7382
[4] orcid: 0000-0001-9536-4947

https://doi.org/10.1016/j.cmpb.2023.107712

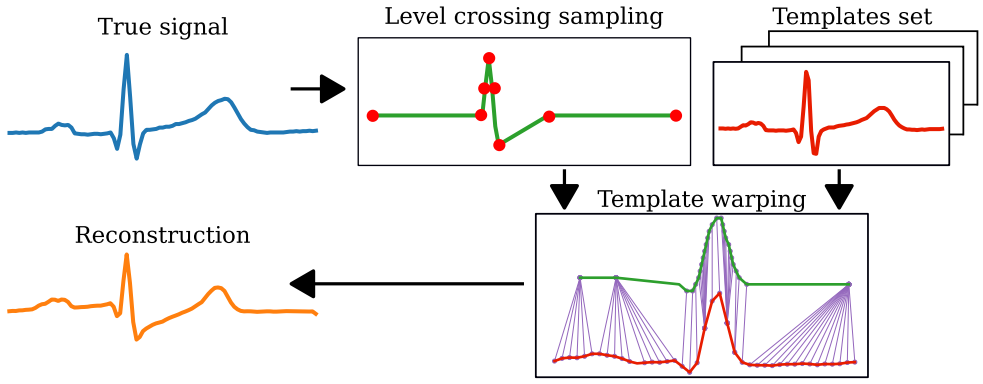

Fig. 1. ECG signal reconstruction processing pipeline: each heartbeat is first sampled with a level crossing ADC, then compared to a set of templates. The most similar template is then warped to better represent the recorded events. In this example, the EB-sampler uses a small number of levels. While not drawing any samples from the P and T waves, we are still able to determine the most appropriate template amongst the templates set to lead the reconstruction, thanks to the morphological distance between the exemplar heartbeats and the drawn events.

Nyquist-Shannon theorem are called uniformly sampled. However, it is possible to define many sampling schemes that do not rely on a uniform spacing between samples. One of the main non-uniform sub-Nyquist sampling approach is event-based (EB) sampling: a data-acquisition strategy that aims to record signal points only when certain events happen in the signal.

In recent years, two main event-based sampling techniques have emerged: 1) level-crossing analysis [8], and 2) polygonal approximation [9], which have shown to be able to greatly reduce the average sampling frequency of a signal up to 90% [9]. However, as the reduction rate increases, the signal fidelity greatly degrades, since all these acquisition approaches are lossy by their very nature.

We depart from the aforementioned sampling methods, considering the acquired events as fiducial points of the recorded signal. We then use these points, together with a set of representative signal templates, to drive a novel approach to signal reconstruction.

Our approach is applicable to signals that are representable by a set of templates, such as the electrocardiogram (ECG) [10]. Indeed, since the ECG is the recording of the electrical polarization of the muscular tissue of the heart [11], only a finite amount of ECG morphologies can physically exist [12]. These morphologies are characterized by the presence, absence, and shape of three main complexes, representing three polarization and depolarization phases of the cardiac muscle: P-wave, QRS complex, and T-wave. Hence, we can consider an ECG recording as the non-exact repetition of a set of patient-specific heartbeats (i.e.: templates), where every repetition has a degree of deformation with respect to a selected template.

We embodied this approach in the processing pipeline shown in Fig. 1, where we reconstruct an ECG signal sampled using level crossing [8]. To achieve this, we leverage the ECG self similarity [13] and a novel customization of the dynamic time warping algorithm (DTW) [14], that we called Information-Injected Differential Dynamic Time Warping (II-DDTW). First, the presented system uniformly acquires a set of heartbeats (templates) representative of the signal. Then, for every EB sampled heartbeat, it uses II-DDTW to both find the best fitting template and optimally deform it so as to pass through all the recorded events. Moreover, whenever the templates set is not anymore representative of the underlying signal, the processing pipeline acquires a new templates set.

The processing pipeline proposed here marries two fundamental requirements in modern wearable medical technology: a low data rate to reduce battery usage, and the need for a morphologically correct signal. The methodology here proposed allows us to delegate signal reconstruction to computationally powerful devices, while greatly reducing the amount of energy an edge device needs to acquire, store, and send data.

We test the developed processing pipeline against the reference MIT-BIH normal sinus rhythm database [15]. This database constrains the variability in the input by considering only normal rhythm situations, allowing us to compare our method with standard resampling techniques, while focusing on the key aspects of signal reconstruction. While more refined strategies can be taken to extend the scope of this work to abnormal cases, they can be seen as incremental improvements to make our methodology more effective in capturing significant templates in those cases. Furthermore, the normal sinus rhythm signal morphology is similar, but for a matter of wave deformation, to many abnormal sinus rhythm [16], directly linking our methodology to correct morphological reconstructions in many abnormal situations such as bradycardia or tachycardia [16, Chapter 13]. The reconstructed signals obtained with our approach are evaluated using three merit figures: 1) an element-wise aggregated distance (percentage root mean square difference), which measures overall signal correctness, 2) a morphological distance (DTW distance), measuring the similarities between the compared heartbeats, and 3) P/T wave delineation accuracy, an application specific measures that shows if ECG relevant structures are present and rightly positioned. These three measures, spanning from signal agnostic to signal specific, give a comprehensive view of the reconstruction accuracy. The results are then compared to the results obtained by reconstructing the signal with three different methods: sample&hold, linear resampling, and spline resampling.

Hence, the main outcomes of our work are:

- The relevance of self-similarity in EB-sampled signals.
- The possibility, for EB-sampled signals, to be represented by a templates set.
- The morphological mapping between templates and EB-signals, through a novel strategy called II-DDTW.

2. Background work

In this section, we first explore the core concepts we leverage in our work. Then, we review the works most similar to ours, highlighting the key differences and the concepts we ourselves took as inspiration.

2.1. Foundation notions

Our methodology relies on three core concepts: 1) self-similarity, 2) event-based sampling, and 3) dynamic time warping (DTW).

Self-similarity: We define a signal to be self-similar when it can be approximately described by a function of a subsection of the signal itself. The term "self-similar" appears in other disciplines and works as a characteristic of a fractal-type mathematical object [17]. The self similarity aspect of the ECG signal is used in [13]. In their work, [13] design a compression system for ECG signals based on multi-scale analysis, where each heartbeat is encoded as the scaling and rotations of prototype heartbeats.

Event-based sampling: Event-based sampling is the action of drawing the value of a measured signal only when it has a predetermined behaviour (an event), using an event-based analog to digital converter (EB-ADC).

To represent a signal using only its samples requires a set of conditions on the sampling function. Most famously, the Nyquist-Shannon theorem [7] imposes a uniform-grid sampling frequency of two times the maximum spectral content of the signal under analysis. Other approaches [18,19] broaden this condition while still achieving a loss-less reconstruction of the sampled function. However, these approaches do not use any contextual information to determine whether a signal section conveys important features or not. Instead, the approach here analyzed defines an event as a logical condition that can be verified based on the signal behavior, and such events are then used to trigger the sampling process.

The two most prominent examples of EB-sampling are level-crossing analysis and polygonal approximation. In level-crossing analysis [8] an event is generated every time the measured physical quantity cross a set of levels. Such operation can either be analog [20], where a sampler detects the crossing of analog-defined levels or digital [8], where the signal is digitally acquired by a standard analog-to-digital converter (ADC) with the addition of a custom logic that defines a set of digital levels and forwards the samples to the main processing unit only when such levels are crossed. Instead, in polygonal approximation [9] the signal is digitally acquired by a standard ADC but with the addition of a custom logic that forwards the samples to the main processing unit only when the error between the uniformly sampled signal and its linear approximation grows bigger than a threshold ε.

Moreover, EB sampling is related to technologies such as Compressed Sensing (CS) [21], which achieve data compression through two signal-agnostic measurement matrices, making the acquired signal sparse. However, contrary to EB-sampling, in CS the length of the sparse representation is signal independent.

Dynamic time warping: DTW [14] is an algorithm that takes as input two vectors, not necessarily with the same dimensionality, and outputs a distance between the two, together with a vector matching sequence. Such sequence, also called edit path, associates every point in the first vector with one or more points of the second vector. This algorithm is of particular interest to this work since it has proven to be effective in the alignment of ECG recordings [10] and because of the interpretability of the output distance as a morphological difference between the two input vectors. Formally, given two vectors v_1 and v_2 with dimensions N and M, the DTW algorithm computes the matrix $D \in \mathbb{R}^{N\times M}$ following Eq. (1):

$$D[i,j] = |v_1[i] - v_2[j]| + min(D[i,j-1], D[i-1,j-1], D[i-1,j]) \quad (1)$$

Where i is an integer that ranges from 2 to N, j is an integer that ranges from 2 to M, $D[i,j]$ is the element of the D matrix at row i and column j, $v_1[i]$ and $v_2[j]$ are the ith and the jth components of the two vectors v_1 and v_2.

The boundary conditions on the D matrix computation vary depending on the priors the user might define. However, they are typically defined as $D[1,1] = 0,\ D[1,j] = D[i,1] = \infty$ [14]. The warping path is computed starting from $P[1] = (N,M)$ and selecting the next point in the path according to Eq. (2):

$$P[l-1] = (i,j)$$
$$P[l] = \underset{i,j,i-1,j-1}{\arg\min}\{(D[i,j-1], D[i-1,j-1], D[i-1,j])\} \quad (2)$$

The process is iteratively repeated until $D[1,1]$ is reached. The list $P[1],\ldots,P[l]$ is then a sequence of tuples that associates each point in v_1 to one or more points in v_2 and vice versa, keeping the order relation between samples and warping the two signals by elongating sections of the two vectors. Finally, $D[N,M]$ is called the DTW distance and can be interpreted as a measurement of morphological distance, as it increases with the path length and with the distances in each matched point. DTW computes the morphological distance between two vectors, and the warping path that minimizes this distance. In this work, we use these properties to compute the pertinence of a template to an EB-sampled beat, and then to match each event.

The general formulation of the DTW algorithm shows some practical issues when dealing with signals without prominent matching features or with missing signal samples (as it is the case for event-based signals). The most relevant problem is the singularity problem [14,22] where multiple points of one vector get (wrongly) associated with one single point of the other. Moreover, the singularity problem exacerbates, in extreme cases, into the information mapping problem, where two non-matching features (where a feature is a distinct set of points not caused by noise) between v_1 and v_2 get mapped together.

Several techniques have been developed to assess this problem like distance matrix step pattern re-definition [23] and differential dynamic time warping [22]. In particular, this last technique consists in changing the distance function presented in Eq. (1) to the one in Eq. (3) to include the difference in derivative instead of in value, thus capturing better a morphology change instead of the specific value.

$$D[i,j] = \left|\frac{v_1[i]-v_1[i-1]}{t_1[i]-t_1[i-1]} - \frac{v_2[j]-v_2[j-1]}{t_2[j]-t_2[j-1]}\right| + min(D[i,j-1], D[i-1,j-1], D[i-1,j])) \quad (3)$$

Where the same notation of Eq. (1) is used, t_1 and t_2 are the two time vectors that describe the sampling instants of the elements in v_1 and v_2 (and therefore have the same dimensions of their respective value vectors). Moreover, we use the first approximation of the derivative as it already fully describes the average rate of variation between two consecutive samples.

Finally, a modern approach we took as an example and point of departure for our work is the technique developed in [24], named Event-Based Dynamic Time Warping (EBDTW). This approach starts by pre-processing the vector, identifying ascending and descending slopes sections[5] in the two vectors. Then, it matches the events between the two vectors before the DTW algorithm. The distance in Eq. (1) is then complemented by the insertion of the constraint that distances are computed only between points that are in the matching slopes.

This last technique is of interest to our application as it introduces the idea of injecting prior information in the computation of the DTW algorithm to improve its performance. However, such a strategy is not a viable solution for our problem, as entire slopes can be removed by the event-based sampling in EB-ECG signals. Moreover, while the technique in [24] is used for matching the same signal recorded by two different methods, in our work we couple a template with any EB-sampled heartbeat. Hence, we can not assume having the same features (in this case slopes) between the two compared signals. To overcome these issues, we introduce

[5] These slopes are called events in [24].

in Section 3 the concept of Information-Injected Differential Dynamic Time Warping (II-DDTW).

2.2. Literature review

The objective of our work is ECG signal reconstruction from a reduced set of key samples. This can be interpreted as the inverse problem to the signal compression task. However, classic decompression techniques [25] are strictly coupled with the relative compression method. Our work, while requiring a specific class of signal, does not need a specific EB-sampling method.

Another related line of study can be identified in ECG representational studies. In [10,26,27] the authors represent full ECG recordings through their most representative heartbeats, clustering each beat so to present a comprehensive view of a patient through key examples. While this approach shares similar intuitions with our work, we use the representative heartbeats for a different purpose, that of templates for heartbeat reconstruction.

Moreover, contrary to the aforementioned works, we determine the representative heartbeats through a graph clustering algorithm [28] called affinity propagation [29] to solve both the uncertainty in the number of clusters and the non-convexity of the formulated clustering problem.

Finally, EB-sampling has seen a significant rise in interest in recent years [8,9,20]. However, signal reconstruction has always been considered a secondary task, assessed using standard interpolation techniques like linear or spline interpolation [30]. Here, we focus on optimizing signal reconstruction in those instances of EB-sampling leading to high sampling reduction factors.

3. Methodology

A high-level view of the proposed methodology is shown in Fig. 2, divided into functional blocks. The reconstruction methodology here presented works on a beat-by-beat basis. Our framework algorithmically selects representative heartbeats as templates. It then uses them to reconstruct event-based ECG acquisitions, warping templates using the acquired events as fiducial points, i.e. forcing the warped template to pass through them.

Section 3.1 details the block in Fig. 2A, where the system EB-samples the ECG signal, mark the QRS complexes, and uses this information to extract EB-heartbeats.

Then, Section 3.2 describes Fig. 2B, where the II-DDTW algorithm (a pattern-matching algorithm) compares each new EB-heartbeat with a set of templates, and selects an optimal template.

Section 3.3 delineates the block in Fig. 2C, where EB-heartbeats are reconstructed based on the warping path defined by the previous block.

Finally, Section 3.4 characterize Fig. 2D, where the templates set is updated in order to keep the set relevant to the current EB-sample signal.

3.1. Acquisition

The starting point for reconstruction is a sequence of samples acquired by a LC-ADC. Samples are represented by the sequence of time-value tuples (t, v): the time of acquisition and the threshold value being crossed.

In Fig. 2A, the system EB-samples the ECG signal using a level-crossing ADC such as the one described in [8]. Then, it marks the QRS complexes and extracts EB-heart-beats. This can be done, as previously shown in [31], by re-implementing the gQRS-detection algorithm [32] to work with EB-sampled signals. Using the QRS timing information, the processing pipeline computes the instantaneous RR interval and uses it to define the heartbeats boundaries. Finally, this processing pipeline section also provides the uniformly sampled heartbeats to the templates set re-computation block, whenever the current set is not anymore representative of the EB-heartbeats. This would require a hybrid approach to sampling, which employs both an LC and a uniform ADC, alternating their power state (on or off) according to the situation, switching between EB sampling and uniform acquisition. We herein assume such scenario, abstracting the hardware implementation details.

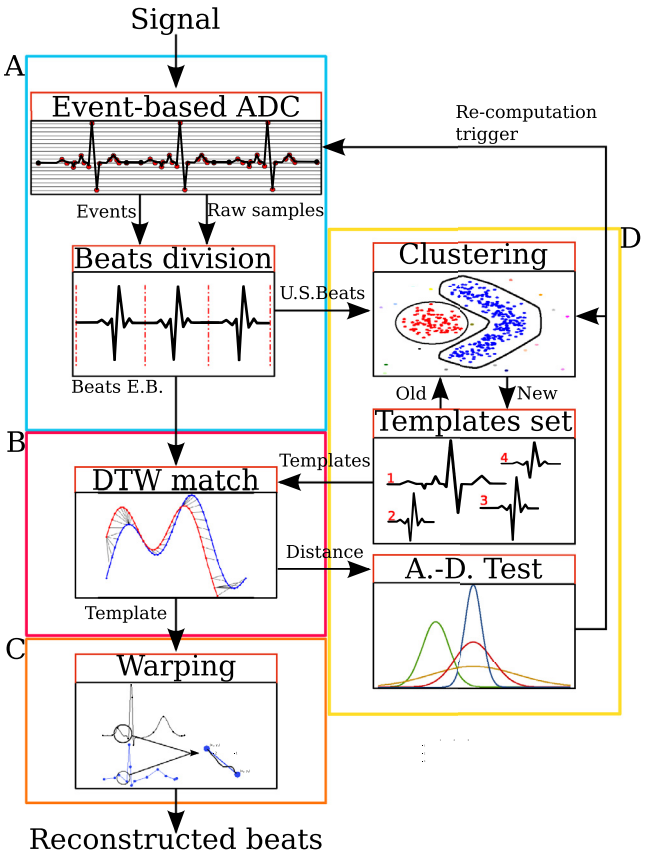

Fig. 2. A bird-eye view of the system described in Section 3. Each labeled box corresponds to specific subsections. A: Acquisition (Section 3.1), B: Differential dynamic time warping with information injection (Section 3.2), C: Template-based reconstruction (Section 3.3), D: Templates computation (Section 3.4).

3.2. Differential dynamic time warping with information injection

We here introduce a novel algorithm, named Information Injected Differential Dynamic Time Warping (II-DDTW), which is an evolution of the DTW algorithm described in Section 2.1. We use this novel approach to find the best fitting template for a EB-sampled heartbeat, and compute the warping parameters for the chosen template.

Information-Injected DDTW. To address the DTW problems mentioned in Section 2.1, first, we opt to use the differential approach to the DTW algorithm (DDTW), expressed in Eq. (3). Then, we re-formulated the DDTW accumulated distance to take into account additional information and use it to guide the warping process. The developed formulation for Information-Injected DDTW (II-DDTW) is shown in Eq. (4), where the t_1, and t_2 terms represent the aforementioned additional information.

$$\begin{aligned} D[i,j] = &(1+\lambda\left|t_1[i]-t_2[j]\right|) \\ &\cdot\left|\frac{v_1[i]-v_1[i-1]}{t_1[i]-t_1[i-1]}-\frac{v_2[j]-v_2[j-1]}{t_2[j]-t_2[j-1]}\right|+ \\ &min(D[i,j-1],D[i-1,j-1],D[i-1,j]) \end{aligned} \tag{4}$$

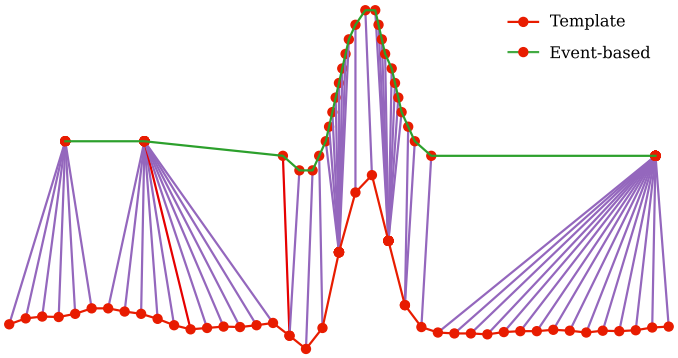

Fig. 3. Example of an EB sampled heartbeat coupled with a template. The warping path is represented here by the purple lines, connecting events with the corresponding points in the template. The red lines exemplify the middle point in the points set associated with the corresponding event. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The motivation of such a formulation can be found in the very nature of our study. EB-sampled signals are composed of data points represented by a tuple: (time, value). We can define a time base also for uniformly sampled signals using the sampling period. The time difference information leads the mapping between events and templates, enforcing a loose time matching requirement.

Moreover, since we can not ensure template and EB-sampled heartbeats to have the same duration, we normalize the timescale for both vectors to 1. We use the differential version of the DTW distance to better capture the changes in morphologies. Finally, we introduce a correction factor λ to modify the behavior of the algorithm, giving more weight to the accumulated distance, as per Eq. (3), or the additional (injected) information, in our case time, resulting in the multiplication factor in Eq. (4).

3.3. Template based reconstruction

Using the definition of Section 2.2, sections of the ECG signal are represented by morphological deformations of templates. Given an EB-heartbeat, the II-DDTW block (Fig. 2B) computes the II-DDTW between the EB signal and all the elements of the templates set. Then, it selects the template with the smallest distance as representative.

To warp the selected representative template, the warping block (Fig. 2C) uses the warping path to match every couple of consecutive events to a set of corresponding points in the template. As shown in Fig. 3, to define the template segment corresponding to the each EB tuple the warping block uses the central half of the points assigned to the events: half of the points associated with the first event, and half of the points associated with the second event. This process produces a set of uniformly sampled segments (coming from the selected template), each associated with a specific segment in the EB sampled heartbeat.

The reconstruction is then performed by deforming each template segment such that the edges of the segment match the two corresponding EB samples. First, the EB segments are shifted such that the first element is (0,0), as described in Eq. (5):

$$\begin{aligned} E_{i_s}[j] &= E_i[j] - E_i[0] \\ T_{i_s}[j] &= T_i[j] - T_i[0] \end{aligned} \tag{5}$$

Where E_{i_s} and T_{i_s} are the shifted event-segment and the template-segment. E_i is the ith event-segment composed of $E_i[1] = (t_i[1], v_i[1])$, and $E_i[2] = (t_i[2], v_i[2])$, $T_i[j]$ is the jth element of the ith template-segment, described by the tuple (t, v).

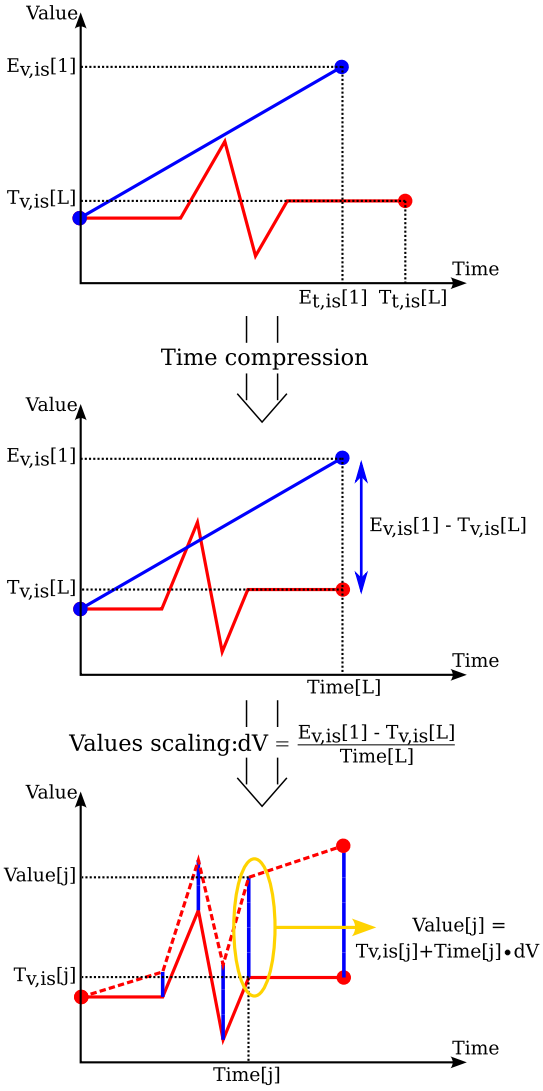

Fig. 4. Exemplification of warping based on II-DDTW. In blue, a segment defined by two events, in red, the corresponding template segment. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The deformation is then performed by computing a warped template segment as in Eq. (6):

$$\begin{aligned} Time[j] &= T_{t,i_s}[j]\frac{E_{t,i_s}[1]}{T_{t,i_s}[L]} \\ Value[j] &= T_{v,i_s}[j] + Time[j]\frac{E_{v,i_s}[1]-T_{v,i_s}[L]}{Time[L]} \end{aligned} \tag{6}$$

Where $T_{t,i_s}[j]$ is the jth time element of the shifted ith template section, $T_{v,i_s}[j]$ is the jth value element of the same template section. The same notation can be seen used for the corresponding shifted event segment, in the form $E_{t,i_s}[j]$ and $E_{v,i_s}[j]$, L is the length of the template section. Finally, $Time[j]$ and $Value[j]$ are the time and value jth component of the newly created resampled heartbeat segment.

This operation is graphically depicted in Fig. 4, aligning the edges of the template and event segments, while smoothly varying the values of the template segments by the same $\Delta v = \frac{E_{v,i_s}[1]-T_{v,i_s}[L]}{Time[L]}$ for each time-unit. The operations showed in Eq. (6) compute a sequence of (t, v) vectors uniformly sampled but not with the same time-base as the original signal and segment-dependent. Finally, the heartbeat is reconstructed by concatenating

all the warped segments (adding the offset subtracted in Eq. (5)) and linearly resampling the data to obtain a uniform time-base.

3.4. Templates computation

The templates computation is divided into five stages: re-computation trigger, U.S. Beats acquisition, clustering, templates filtering, and set re-computation.

Templates set re-computation trigger. After a new templates set is computed, the template computation block (Fig. 2D) collects a reference set of 400 DTW distances from the results of Section 3.2, assuming that, in this time span, the templates set is representative of the patient heartbeat morphologies. Hence, the distance set acquired is used as a reference to estimate the distribution of the DTW distance of a coherent templates set. After the initial training time, the template computation block acquires a test batch of distances every 60 s and compare it with the reference distribution using the A.D. test [33]. This test returns a critical value bounded by the probability that the test sequence came from the same distribution of the reference sequence. If such probability is lower than 0.05 for two consecutive times, then a new templates set is computed. This double confirmation is required to avoid triggering the templates set re-computation if the signal in one time window was affected by external factors that do not modify the beats morphology but might modify the recording (e.g., motion noise). The recomputation trigger presented here assumes only normal sinus rhythms that vary slowly over time. Although more precise testing can be used to capture single abnormal heartbeats, we opted for a simpler formulation of our processing pipeline. The refinement of our strategy to deal with abnormal heartbeats is a natural extension of this work.

U.S. Beats acquisition. When a new templates set is needed, the first step is to start the signal acquisition at a uniform sampling rate. The uniformly sampled signal is then divided by the beats-division algorithm in U.S. beats. The first time the templates set is computed, the U.S. beats acquisition is performed for a longer time window (compared to templates set re-computation during normal functioning). Each heartbeat is then min-max normalized and used as input for the clustering algorithm (Fig. 2).

On an opposite note, during the re-computation stage, the acquisition time is shorter. This short acquisition time is required since this operation is triggered when the morphology of the current heartbeats do not resemble the ones contained in the already present templates set. The processing pipeline, hence, needs to quickly adapt to signal changes and a search-space small enough to consider the new morphologies as separated clusters and not outliers of already present and dominant ones.

Self-organizing clustering. The templates set is computed by clustering the acquired heartbeats and selecting the U.S. beats closest to the centroids. The choice of the clustering method is then driven by three main facts: 1) the number of clusters can not be determined beforehand. While it is true that a heartbeat can assume only a finite number of morphologies [34], this would require the U.S. beats set under analysis to be statistically complete and without biases, and this can not be ensured for any subset of collected beats. 2) We can not ensure the clusters are convex for any given metric. Also, since our algorithm iteratively computes local clusters, their distribution could change between iterations, making convex conversions techniques such as the kernel trick [35] not viable. 3) The clusters centroids need to be elements of the clustered set, since we can not ensure a heartbeat defined as an aggregation of examples to be physically coherent.

These constraints are satisfied by graph-based clustering methods [28]. Hence, our framework uses a self-organizing, graph-based clustering algorithm called affinity propagation [29]. This algorithm requires two sets of parameters: graph distances and preferences. The preferences vector is equivalent to a statistical prior that defines the likelihood of a data point to be a centroid. In its mathematical formulation, affinity propagation can be exemplified as follows: given a set $X = \{x_i\}_{i=1}^{N}$ of vectors, we define the affinity matrix $S \in \mathbb{R}^{N\times N}$ where each element $S[i, j]$ is defined as a distance function between x_i and x_j. We can then define two matrices: the responsibility matrix $R \in \mathbb{R}^{N\times N}$, which quantifies how well suited the vector x_i is as an example for x_j, relative to other candidates in X, and the availability matrix $A \in \mathbb{R}^{N\times N}$, which represents how appropriate it would be for each vector x_i to pick another vector x_j as an example, taking into account the responsibility of x_j as an example for the remaining points. Both matrices are initialized to 0. We then compute the update of R using Eq. (7).

$$R[i, j] = S[i, j] - max(A[i, l] + S[i, l]), l \neq j \tag{7}$$

The availability matrix is then updated following Eq. (8)

$$\begin{aligned} A[i, j] &= min(0, R[j, j] + \sum_{l\notin(i,j)} max(R[l, j])), i \neq j, \\ A[i, i] &= \sum_{l\neq i} max(R[l, i]) \end{aligned} \tag{8}$$

These operations are executed iteratively until convergence or for a predefined number of iterations. Cluster centroids are then defined as elements in X such that $R[i, i] + A[i, i] > 0$

Our system uses the DTW measurement as the graph distance to capture the concept of morphological similarity between the dataset points. We then use the same preference value for each dataset point since we want the representative heartbeats to be emerging from the graph itself, without any external bias.

Centroids filtering. The centroids and relative clusters are filtered to satisfy two criteria: cluster dominance and the smallest possible signal-to-noise ratio (SNR). The first filter removes clusters that do not contain enough heartbeats. This threshold is 5% of the total number of points in the dataset. In this scenario, the outliers to the clustering algorithms are the noisy portion of the signal. The second filter searches in each of the not discarded clusters a data point that is both the nearest possible to the centroid and with an SNR bigger than 17dB (50 in the linear scale). The values used for these two filters were empirically computed with the objective of being very selective, and yet always allowing a non-zero amount of cluster and centroids in the small signal section analyzed.

To calculate the SNR value, the templates set computation stage first estimate the true signal and then the over-imposed noise. Then, assuming the noise to be a zero-median signal, it divide the signal from the noise applying a median filter to the acquired signal. This approach, however, requires the median filter to be as long as possible and short enough to not classify main signal features as noise. Given these conditions, the median filter time-span is set to 24 ms, based on the average duration of the QRS complex in an average heartbeat [11].

Templates set update. To create an updated version of the templates set, we compare the old and the newly computed templates. Algorithm 1 details the templates set update process. The *update_templates_set* procedure inputs are the old templates set (*old_t*, list of templates) and the newly computed clusters (*new_t*, a data structure holding the new templates, clusters centroids, and distances between all cluster elements and relative centroid). Moreover, the *dist_from_centroid* (lines 7 and 16), computes the DTW distance between a template and a centroid.

In Algorithm 1, lines from 4 to 11 check if old templates are represented in the new clusters: first, line 5 fetches the nearest cluster to the old template under analysis, then, line 6 defines the threshold below which the old template is considered part of the new cluster. This threshold is the average DTW distance plus one standard deviation between the centroid and the elements of the cluster. If the old template is considered inside its nearest cluster,

Algorithm 1 Templates set update algorithm.

```
1:  procedure update_templates_set(old_t, new_t)
2:      near_templates ← Dict(List)
3:      new_set ← List
4:      for old ∈ old_t do
5:          near_new = get_nearest(old, new_t)
6:          threshold = cumpute_th(near_new.all_dists())
7:          d = dist_from_centroid(old, near_new)
8:          if d ≤ threshold then
9:              near_templates[near_new.id] ← old
10:         else
11:             new_set ← old                          ▷ Keep old
12:     for new ∈ new_t do
13:         if new.id ∈ near_templates.id then
14:             old_candidates = near_templates[new.id]
15:             near_old = get_nearest(new, old_candidates)
16:             d = dist_from_centroid(near_old, n)
17:             if d ≤ new.dist then
18:                 new_set ← near_old                 ▷ Keep old
19:             else
20:                 new_set ← new                      ▷ Update old
21:         else
22:             new_set ← new                          ▷ Insert new
        return new_set
```

it is added to the list of possible representatives (line 9). Otherwise, it is kept (line 11) since it represents a morphology not captured by the current clustering iteration. Lines 12 to 22 check if a new cluster is either a) represented by an old template (line 18), b) represented by the newly identified one (line 20), or c) if it represents a new morphology (line 22).

3.5. Processing pipeline composition

The run-time behaviour of the framework depicted in Fig. 2 and detailed in the previous sections can be summarized as follows:

1. As the pipeline execution starts, no templates set is defined. The re-computation trigger initializes the templates acquisition block. This computation lasts for a defined but parametric amount of time.
2. The beats division block divides the uniformly sampled signal into heartbeats. Each heartbeat is sent to the clustering algorithm block.
3. The centroids computed by the clustering algorithm get filtered as described in Section 3.4. As no previous templates set exist, the obtained centroids are not compared to any previous template.
4. When the templates set computation is ready, the system starts the event-based signal acquisition.
5. The EB-gQRS [31] algorithm finds the QRS complexes in the event-based signal and, consequently, subdivide the EB signal in EB heartbeats.
6. The II-DDTW algorithm in the DTW matching block matches any EB heartbeat to every template in the temples set. It then selects the template with the minimum distance from the analyzed EB heartbeat and marks it as representative of this heartbeat.
7. For each EB heartbeat, the DTW matching block matches and warps template segments accordingly to the recorded events and the warping path obtained from the II-DDTW algorithm.
8. Each warped segment is then recomposed together by the template warp block to obtain the reconstructed heartbeat.
9. In order to find if a new templates set is needed, the A.D. Test block saves back the warping distances between the EB heartbeats and the selected representative template.
10. After a new templates set is computed, the A.D. Test block acquires a vector of DTW distances as reference, assuming that a newly computed template is representative of the heartbeats generated immediately after the set computation.
11. After defining the reference distance vector, the A.D. Test block acquires a new test vector and checks if it comes from the same probability distribution.
12. In case the A.D. Test fails ($p - value \leq 0.05$) for two consecutive times the re-computation trigger signal starts a new templates set acquisition.
13. Once a new templates set is computed, the clustering block compares it with the previous set and merges the two as seen in Section 3.4.

An implementation of this processing pipeline has been published as open-source[6], using Python as the primary language, with C implementations for the most compute-intensive routines such as II-DTW. The implementation is highly modular and parametric, allowing to freely vary the acquisition time for templates set computation, the lengths of the reference and test distance vectors, the clustering and centroids filtering parameters, the beats-subdivision window timing, and the number of points relevant for each template-segment/event bounding. The parameterization capability makes the algorithm suited for patient-specific tuning and application on different, non-ECG-related, self-similar signals that can be explored in future studies.

3.6. Computational complexity

Finalizing our methodology characterization, we describe its computational complexity.

Following the description in Section 3.5, we can see that, considering each heartbeat as input, all the forward operations (i.e.: not the templates set re-computation), execute in linear time: when executing the forward operations, for each functional block seen in Fig. 2, the output is solely dependent on the current heartbeat, hence, increasing the number of heartbeats leads to a directly proportional increase in computation time. Moreover, the complexity of the DTW algorithm is $O(N \times T)$ [29] where N is the dimension of the first input and T is the dimension of the second input. II-DDTW does not change the complexity of the algorithm, since the main difference between the two methods consists in using Eq. (4) instead of Eq. (1), which does not change the complexity of the similarity matrix definition operation (i.e., the operation with $O(NXT)$ complexity). However, their execution time is fixed when the input size is fully determined. Although in our environment this is not strictly true, the average sizes of the input parameters are well-defined and do not change depending on the number of heartbeats processed. The deployed A.D. test has a linear complexity and is executed every 60 s. Hence, the number of times the re-computation trigger is set (i.e.: when the obtained critical value is smaller than 0.05), must be an integer fraction of the number of times the A.D. test was executed. Hence, the number of times the feedback loop is triggered depends linearly on the number of heartbeats processed. Finally, we notice that the templates set re-computation, and especially the affinity propagation algorithm, behave quadratically with respect to the number of inputs. However, the number of heartbeats used for clustering is determined by the constant acquisition time used for templates set re-computation. Since this time span is fixed, the templates-set feedback loop executes in constant time.

[6] https://gitlab.epfl.ch/zanoli/EB_ECG_Smart_Resampler

From this analysis, we can conclude that the processing pipeline developed has a linear complexity with respect to the number of heartbeats analyzed.

However, we notice that while we can define the algorithmic complexity of the full processing pipeline, this complexity does not characterize a potential hardware deployment, since in a real-world scenario the proposed methodology would be split between an edge device and a back-end server, to optimize edge energy consumption.

4. Experimental setup

4.1. Data and methods

To validate our work, we test it against the MIT-BIH Normal Sinus Rhythm database [32]. This database comprises 18 long-term (approximately one day) ECG recordings of subjects referred to the Arrhythmia Laboratory at Boston's Beth Israel Hospital (now the Beth Israel Deaconess Medical Center). The total number of heartbeats analyzed is approximately 1.8 million. The ECG recordings are sampled at a frequency of 128 Hz.

To decouple the performances of QRS detection from our algorithm, we relied on the ground-truth annotations in the selected dataset. This decision allows us to focus on the performance evaluation of the work developed here (the processing pipeline), effectively uncoupling our results from the performance of any QRS detection algorithm.

Using the QRS time locations, the processing pipeline computes the instantaneous RR interval and uses it to define the heartbeats window. The window boundaries for each heartbeat is defined by the time of the QRS complex minus 40% and plus 60% of the RR interval. Then, the acquisition block checks if any event is present at the boundaries time and, if not, it insert a synthetic event with a zero value. This approach is consequence of the boundary timing, as it is computed so to fall after a T-wave and prior to a P-wave, where no cardiac activity is present.

In our experiments, the regularizer λ in Eq. (4) has been set to 1. An empirical exploration of this meta-parameter led to the conclusion that, for our specific problem, a different choice of λ has a very weak influence on the reconstruction result.

Finally, the initial long-term U.S. beats acquisition for the first templates set computation is set to 3 min, while the shorter time span for template-set re-computation is set to 40 s. These values have been empirically selected: a shorter acquisition time leads to less representative templates (undermining the results), and a longer time span do not lead to significant improvements.

4.2. Data rate reduction metric

One important outcome to consider in our experiments is the potential for energy savings of our methodology. While we presented, in the previous sections, a fully algorithmically defined pipeline, we can use its functional block formulation to define an energy-efficient strategy of implementation in an edge-to-cloud scenario. The implementation strategy we envision for our methodology uses an edge device to solely acquire and send data, using the smallest amount possible of processing power, while delegating the more difficult and energy-intensive tasks to a more powerful back-end. This creates a clear division of the tasks. In particular:

- The edge device: Acquires templates, acquires EB Signal, computes templates coherence with respect to EB Signal, sends data to back-end.
- The back-end: receives EB heartbeats, receives templates, computes optimal II-DDTW reconstruction.

The proposed processing pipeline makes use of the II-DDTW distance to evaluate the coherence of the templates. This increases the energy computation on the edge device, as this operation is repeated every new EB-heartbeat is acquired. However, this effect is counterbalanced by the smaller number of events acquired and transmitted. Moreover, template recomputation also poses a significant processing overhead in the edge device application. While this is a limitation of our methodology, many different strategies exist to accomplish templates coherence evaluation, and the impact of templates set recomputation is largely dependent on the average time between each recomputation. To evaluate the potential energy savings in the edge device, we compute the average reduction in data rate. While a complete analysis would require the energy evaluation of the edge device computation, this is device specific and out of this work scope. Moreover, the energy for data processing is orders of magnitude lower than both the energy required for memory retention and RF transmission. As an example, we can notice how Bluetooth low energy consumes, at least, 0.1W [36], while the consumption of a low power microcontroller unit while in active state is usually lower than 10μW [5]. To evaluate the data rate savings, we then use Eq. (9).

$$R = 1 - \frac{2p(1-SRF)Tf + (1-p)Tf}{Tf} = p(2SRF - 1) \tag{9}$$

where p is the percentage of time the system is acquiring an EB signal, T is the total time length of the recording, f is the uniformly sampled signal frequency, and SRF is the sampling reduction factor achieved by the EB sampling, defined as $SRF = \frac{Tot.Points - EB\_points}{Tot.Points}$: the ratio between the points discarded by the EB-sampler and the number of points in the uniformly sampled signal. Analyzing Eq. (9), the term $p(1-SRF)Tf$ is the number of points sampled by the EB-ADC, and the factor 2 is due to the need to send information about time and value, which is not strictly needed for uniformly sampled signals. Finally, while Eq. (9) does not include the energy needed to transmit the templates, together with the signal. However, we can notice that the templates are among the uniformly sampled heartbeats, hence, it is always possible to obtain the template-set relative to each EB-sampled heartbeat by either using the clustering algorithm in the back-end a second time, or by adding a single bit to every uniformly acquired heartbeats to identify which one is considered representative. Considering an average number of 94 points per heartbeat (acquiring at a sampling rate of 125Hz time the average heart rate in seconds: 0.75 bps, or 80 bpm), each point being represented by 10 bits, we can state that Eq. (9) is a good approximation for data rate saving.

4.3. Evaluation metrics

The performances of the hereby proposed processing pipeline are evaluated using three merit figures: 1) percentage root mean square difference (PRD) [37], 2) Dynamic time warping distance, and 3) P and T wave delineation F_1 score. These three metrics complement each other, depicting a comprehensive view of the strengths and weaknesses of our approach, as discussed in the following.

The PRD is a normalized distance between two vectors, computed as in Eq. (10), where x_{org} and x_{rec} are the samples of the original and reconstructed signal:

$$PRD = 100 \cdot \sqrt{\frac{\sum_{i=1}^{n}(x_{org}[i] - x_{rec}[i])^2}{\sum_{i=1}^{n} x_{org}[i]^2}} \tag{10}$$

The next metric explored broadens the definition of distance to a non-strictly mathematical one [38] using the original DTW warping distance as a merit figure (i.e., $D_{N,M}$ in Eq. (1)). This is the cumulative ℓ_1 distance between elements in the optimal warping

path between a reconstructed and the corresponding true heartbeat.

The final metric we analyze is wave delineation. A normal ECG heartbeat is composed of three main waves: P, T, and QRS. The QRS complex detection is, in this instance, a trivial task: it is a fundamental part of the processing pipeline and the clearest feature of an ECG signal. This feature is, hence, always present and correctly positioned. Conversely, the presence or absence of the P and T waves, alongside with their correct positioning, is a strong marker of well-reconstructed heartbeats. To evaluate the wave presence/correctness, we first use the delineation tool `ECGPUWAVE` included in the `wfdb` software package [32]. This tool identifies the P and T waves in a record. Then, we use the `bxb` tool, also contained in the `wfdb` software package, to find if the waves found in the original signal coincide with the waves found in the reconstructed heartbeats. Using the standards given by the `bxb` software, two waves are considered matching if their labels are, at most, 0.15 s apart. This measure is numerically evaluated measuring the correctly detected waves (*True Positive*, *TP*), the waves wrongly inserted by our approach (*False Positive*, *FP*), and the waves our approach did not manage to recover (*False Negative*, *FN*). Using this data, we compute the sensitivity score $S = \frac{TP}{TP+FN}$, and the positive predictive value $PPV = \frac{TP}{TP+FP}$. Sensitivity measures the percentage of waves in the original signal correctly reconstructed and well-positioned, while the positive predictive value *PPV* measures the probability of a detected (reconstructed) wave to be present in the original signal and not be wrongly inserted. Finally, we compute the $F_1 = 2\frac{S \cdot PPV}{S+PPV}$ score: the harmonic mean between *S* and *PPV*. Since the labeling in the original dataset and in the reconstructed signal are computed using the `ECGPUWAVE`, the obtained F_1 score measures the difference when using a delineation algorithm on a uniformly sampled signal rather than on a reconstructed one.

While the PRD (which is a sum of element-wise distances) represents well the difference in values between two vectors, it also has intuition pitfalls. For example, both a constant signal and a slightly skewed but overall correct reconstructed heartbeat can result in the same PRD.

The DTW distance captures the morphological distance of our results from the true signal, making this operation well suited for evaluating a reconstruction operation. Still, this approach can be misleading, mainly in discerning how a feature is better represented. For example, when approximating a quadratic function, a 2-pieces wise linear function might present a smaller DTW than a quadratic function well centered but with slightly higher values.

The PRD and DTW metrics give us an intuition on the correctness of the values of a reconstructed sequence, while the wave identification marks the presence of the feature of clinical interest. In summary, the correct positioning of such features is insufficient to deem the results correct, but a combination of correct wave positioning, standard DTW, and PRD values denotes a reconstruction where the main ECG elements are correctly placed, with similar values and morphology to the original waves.

4.4. Baselines

Alongside the developed pipeline results, we also compute the metrics mentioned above for the EB signal reconstructed through three standard resampling techniques.

1. Sample-and-hold method, where we hold the last recorded value in-between events. This technique represents the complete set of information obtained from the level-crossing technique sampling, as we have no means to know intermediate samples.

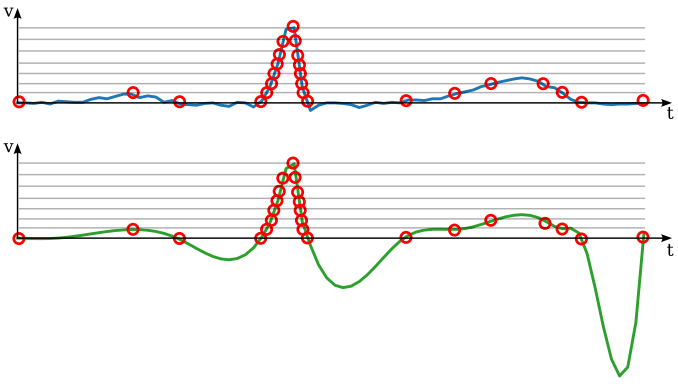

Fig. 5. Spline interpolation of an event-based acquired heartbeat, using an LC-ADC with 6 bits of dynamic range. The non-uniform time distance between samples forces the spline polynomial to be ill conditioned, exhibiting erratic behavior at the end of the signal or at the edges of the QRS complex.

2. Linear interpolation. This technique assumes a constant derivative between two consecutive points.
3. Quadratic spline interpolation [39]. This technique ensures a smooth interpolation. We use 0-valued derivative as a boundary condition. Higher-order spline interpolations were tested as well. However, in these cases the reconstruction suffered unstable oscillations, especially in Q and S waves reconstruction, due to the intrinsic non-linear sparsity of the EB-sampled ECG signal.

All the obtained results are evaluated against the EB sampling level, described by the number of bits used for the level-crossing algorithm (the number of levels being equal to 2^B, where *B* is the number of bits). However, this measurement does not directly represent the effect of the sampling on the signal. We tackle this problem by computing a compression figure called sampling reduction factor (SRF) for each EB sampling level. This metric is the ratio of discarded samples with respect to the total number of samples in the uniformly sampled signal.

To compute all the results mentioned above, we apply the developed processing pipeline to every heartbeat in every record in the dataset. We then compute each metric for each resulting reconstructed beat. Then, we re-evaluate the same metrics using the chosen resampling techniques. Finally, we derive the statistical distributions for both methods and metrics.

5. Results and discussion

The analysis of the spline interpolation results shows, in the distance-based metrics, an error at least one order of magnitude higher than the other resampling techniques. As illustrated in Fig. 5, the reason behind this behavior is the sparse nature of points in the initial and final section of every heartbeat together with high derivative values where the data-point density increases near the QRS complex. This behavior causes the polynomial approximation to be non-representative of the true underlying signal. This problem resides in the very nature of level-crossing sampling, making the spline interpolation a non-effective interpolation method for level-crossing ADCs. The results about spline interpolation are hence not discussed in the remaining part of this section while still being present in the numerical tabulated results.

5.1. Data rate reduction

Figure 6 shows the average SRF value, computed by EB-sampling all the signals in the dataset, for an increasing number of bits in the level-crossing EB-ADC. A negative SRF means that the number of points is higher than in uniformly sampled signals.

ADC bits number	3	4	5
T#	4.73 ± 2.53	5.22 ± 2.55	5.48 ± 2.80
TBT	18.3 ± 16.6	21.0 ± 19.8	20.0 ± 18.3

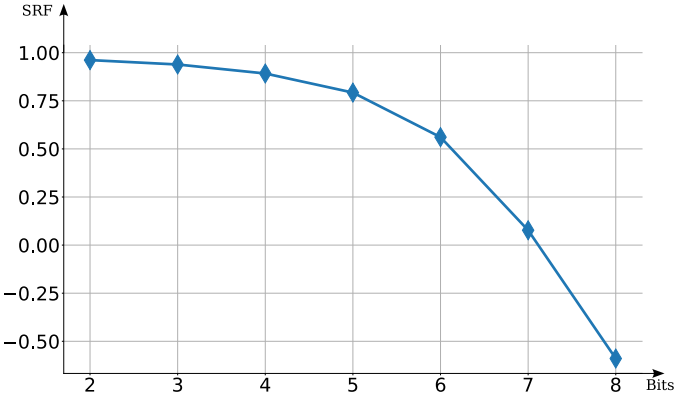

Fig. 6. Sampling reduction factor vs. LC-ADC BIT numbers. Negative values represent a higher number of points with respect to uniform sampling.

Table 1
SRF, Percentage of EB-heartbeat (p), data rate reduction (R).

ADC bits number	3	4	5
SRF	0.935	0.887	0.785
p	0.961	0.967	0.965
R	0.836	0.748	0.550

Table 2
Average number of templates needed for the reconstruction (T#), and average time between templates set re-computation (TBT, in minutes).

ADC bits number	3	4	5
T#	4.73 ± 2.53	5.22 ± 2.55	5.48 ± 2.80
TBT	18.3 ± 16.6	21.0 ± 19.8	20.0 ± 18.3

We restrict the analysis of our results to the 3,4, and 5 bits LC-ADCs, for two reasons: first, while using a 2-bits LC-ADC does not significantly lower the SRF, it fails to detect a single event in each heartbeat most of the times. This leads to results that are disconnected from any signal measurements. Second, the results show only a marginal improvement for LC-ADCs with more than 5 bits, while the results obtained for higher number of bits ($Bits \geq 7$) are indistinguishable from the uniform sampling approach. We then deepen our results analysis, observing in detail 4-bit LC-ADC performances as we notice it to achieve a high SRF while obtaining overall good results.

To obtain the data rate reduction described in Eq. (9), we measure the percentage of heartbeats acquired using EB-Sampling. Table 1 presents the SRF, percentage of EB-heartbeat, and overall data rate reduction. Moreover, as part of characterization of the described methodology, we also report in Table 2 the average number of templates needed for the reconstruction and the average time between templates set re-computation.

5.2. Percentage root mean square difference

Figure 7 presents the distribution of the PRD distance of the three considered reconstruction techniques, using an LC-ADC with 3, 4, and 5 bits of dynamic range. The enclosed box spans from the 25th to the 75th percentiles, the horizontal central line is the median, and the thin outreaching lines enclose the 5-to-95 percentile. Using this figure, we can derive two interesting points: 1) the flat

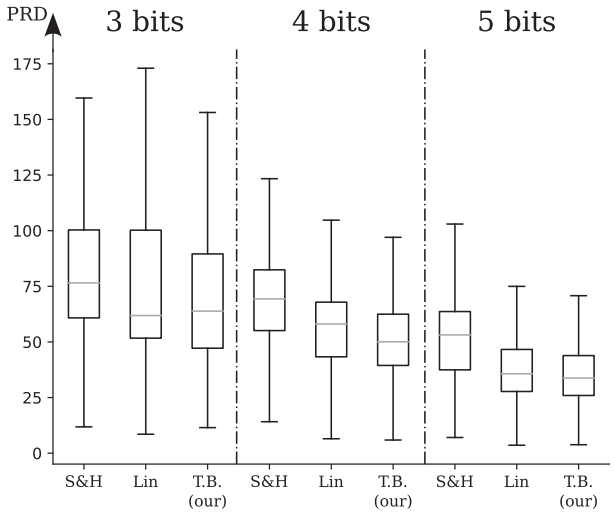

Fig. 7. PRD distance distribution against LC-DC bit number and resampling/reconstruction method, using the number of bits discussed in Section 5.1. S&H: Sample&hold resampling, Lin: Linear resampling, T.B. (our): template-based reconstruction (our method).

Table 3
Average and standard deviation PRD for different resampling/reconstruction techniques and increasing EB-ADC bits.

ADC bits number	3	4	5
Spline interpolation	989 ± 2993	681 ± 1505	552 ± 990
Sample & Hold	79.2 ± 25.0	72.2 ± 26.8	52.3 ± 16.9
Linear interpolation	71.3 ± 27.1	56.8 ± 16.1	37.9 ± 14.3
Template based (ours)	70.5 ± 32.0	52.7 ± 19.8	36.0 ± 14.2

interpolation is consistently outperformed by either the linear interpolation or the II-DTW method, 2) the II-DTW method is comparable to the linear interpolation, outperforming it when applied to a 4-bits LC-ADC.

The first point is the result of the step-defined function the sample&hold method creates. This behavior causes, in fast varying signals, an increasing difference between the constant held value and the true underlying signal. The second point shows us that our technique is as valid as linear interpolation when we only consider samples magnitude. The specific numerical values of the average PRD and standard deviation are shown in Table 3.

Figure 8 shows the PRD value distribution of the linear resample method and the II-DTW reconstruction using a 4-bits LC-ADC. The multi-modal distribution shown in the figure is caused by the different magnitudes of P and T waves. A small number of bits in the LC-ADC causes the distance between sampled levels to be sometime higher than the magnitude of the waves we desire to sample, with several possible behaviours: 1) P-wave is sampled, T-wave not, 2) P-wave is not sampled, T-wave is, 3) both P and T-waves are sampled, 4) both P and T-waves are not sampled. These conditions are less representative as the number of bits increases (and the span between ADC values diminishes). Our methodology significantly mitigates this effect, explaining the better performances displayed in Fig. 7 for LC-ADC with 4 bits of dynamic range.

5.3. Dynamic time warping pseudo-edit distance

Figure 9 presents the distribution of the DTW distance of the three considered reconstruction techniques, using an LC-ADC with 3, 4, and 5 bits of dynamic range. We observe that our proposed technique shows a consistently lower error for all choices of LC-ADC dynamic range.

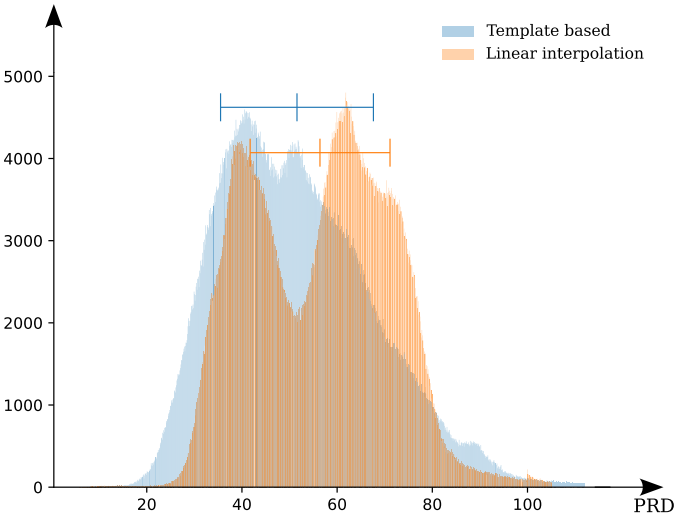

Fig. 8. PRD distribution using an LC-ADC with 4 bits of dynamic range with highlighted average and standard deviation (horizontal bounded lines). Orange is Linear interpolation while blue is our proposed template based technique. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

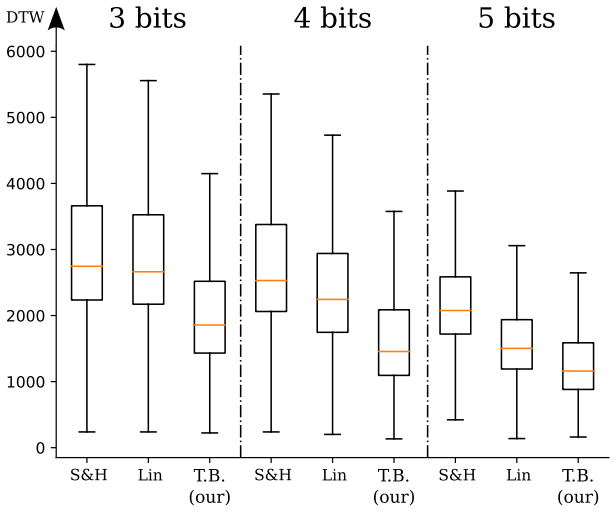

Fig. 9. DTW distance distribution against LC-DC bit number and resampling/reconstruction method, using the number of bits discussed in Section 5.1. S&H: Sample&hold resampling, Lin: Linear resampling, T.B. (our): template-based reconstruction (our method).

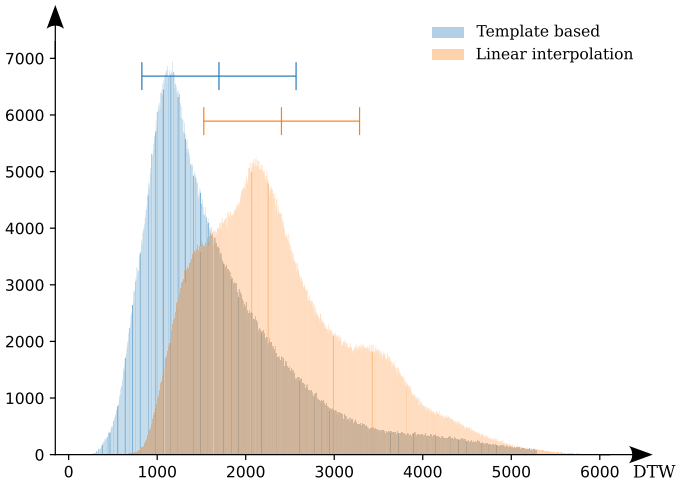

Fig. 10. DTW distribution using an LC-ADC with 4 bits of dynamic range with highlighted average and standard deviation (horizontal bounded lines). Orange is Linear interpolation, while blue is our proposed template-based technique. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

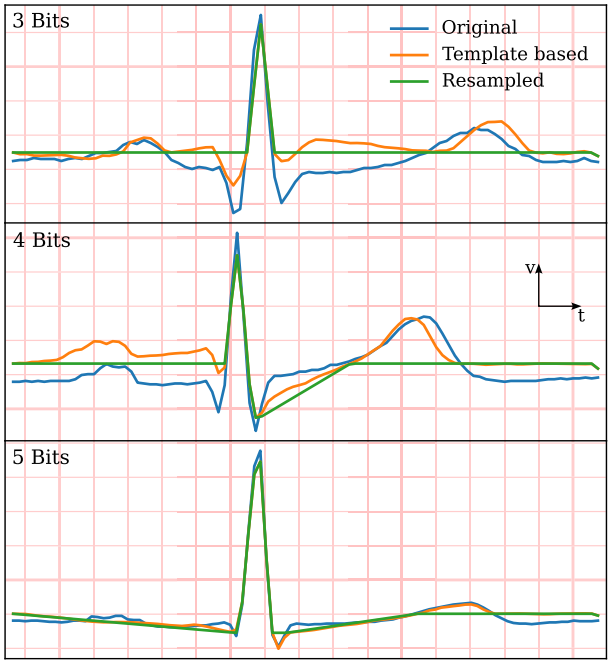

Fig. 11. Heartbeat reconstruction associated with the 50*th* DTW distance percentile, which includes an LC-ADC using 3, 4, and 5 bits of dynamic range.

Figure 10 shows the detailed DTW distribution for the linear interpolation technique and our warping method for an LC-ADC with 4 bits of dynamic range. Here we can see differences in the distribution type between the two methods, with our warping reconstruction exhibiting a unimodal distribution while the linear interpolation method exhibits a composite behavior. These results can be explained by observing Fig. 11, where we select, for all the analyzed levels, the heartbeat reconstruction whose DTW distance corresponds to the 50th percentile distance in the DTW distribution, behaving similar to what we would observe in an average scenario. The self-similarity prior knowledge allows our method to recover significant features also when no points are sampled in the section of interest, while the linear resampling technique is not able to recover any information that has not been recorded.

Finally, Table 4 shows the average and standard deviation of the different reconstruction methods for the selected LC-ADC dynamic ranges. While showing an advantage of our method over classical resampling, these results highlight the accomplishment of the main objective of our work: a good morphological representation of an EB-sampled signal.

5.4. Delineation results

Wave delineation is one of the main objectives in ECG signal analysis. This task consists in detecting the presence, position, and characteristics of P and T waves, alongside the QRS complex. Here, we focus on a sub-task of the wave delineation problem: p and T waves detection.

Table 5 shows the delineation results for the proposed resampling methods and our technique, in terms of F_1 score. We can observe that our method outperforms any resampling technique for any selected LC-ADC range. This is due to the usage of templates

Table 4
Average and standard deviation DTW for different resampling/reconstruction techniques and increasing EB-ADC bits. The results are here scaled down by a factor 10^3.

ADC bits number	3	4	5
Spline interpolation	38.4 ± 36.6	23.0 ± 24.0	15.2 ± 19.2
Sample & Hold	3.05 ± 2.07	2.77 ± 1.37	2.22 ± 0.90
Linear interpolation	2.94 ± 1.70	2.42 ± 1.14	1.62 ± 0.71
Template based(ours)	2.16 ± 1.69	1.74 ± 1.19	1.32 ± 0.78

Table 5
P and T wave delineation results F_1 score for different resampling/reconstruction techniques and increasing EB-ADC bits.

ADC bits number	3	4	5
Spline interpolation - P	0.111	0.237	0.351
Spline interpolation - T	0.655	0.598	0.620
Sample & Hold - P	0.009	0.051	0.101
Sample & Hold - T	0.231	0.394	0.697
Linear interpolation - P	0.007	0.039	0.114
Linear interpolation - T	0.231	0.353	0.617
Template based (ours) - P	0.646	0.696	0.699
Template based (ours) - T	0.814	0.852	0.870

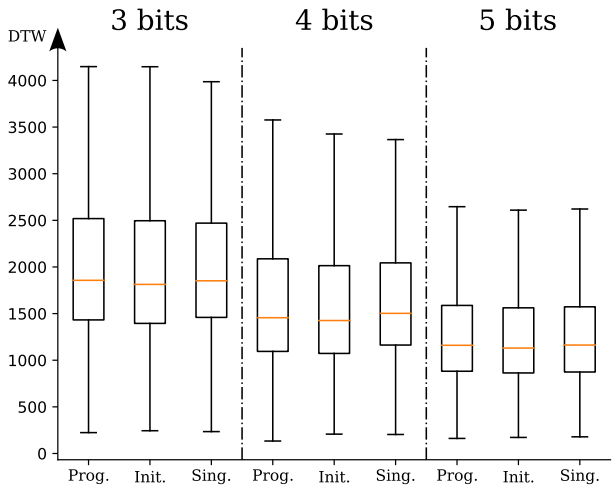

Fig. 12. DTW distance distribution against LC-DC bit number and Templates acquisition type, using the number of bits discussed in Section 5.1. Prog.: Progressive templates acquisition, Init.: Multiple initial templates, Sing. : Single initial templates.

that, if correctly chosen by the processing pipeline, contain similar P and T waves to the true signal, as shown in Fig. 11. The presented waves, warped accordingly to the recorded events, closely resemble the original signal, making the delineation more effective than when applied to a resampled signal.

Finally, we noted that the increase in F_1 score is mainly caused by an increase in the sensitivity score S, and only marginally thanks to a bigger positive predictivity PPV value. To exemplify this point, we observe P wave delineation, using a 4-bits LC-ADC, for linear resampling and templates-based reconstruction. The linear resampling achieve: $S = 0.019$, and $PPV = 0.630$, while using our method we measure: $S = 0.566$, and $PPV = 0.725$.

These results represent the core argument and contribution of our study. The DTW and PRD distances capture an overall improvement achieved by our method over classical resampling. However, they fail to reflect how this improvement is achieved or how it might be helpful for any task. The F_1 score in wave delineation shows that the improvements are caused by an effective reconstruction of the otherwise removed fundamental bio marker of the ECG signal. Moreover, we do not only show an effective reconstruction of the P and T waves but also their correct positioning with respect to the ground truth signal.

5.5. Results discussion

We now discuss how the three metrics analyzed, and their relative improvements, relate to each other. For the sake of brevity, we are going to compare our methodology only with the best-performing approximation (linear interpolation), but similar arguments can be made for each of the considered standard resampling techniques.

First, we must divide the three behaviours observed when evaluating the three described metrics. In delineation, our methodology outperforms linear resampling greatly. Furthermore, the performance gain for the PRD metric achieves an improvement over linear resampling between 1% and 7% for the cases in Table 1. Finally, the performance of our methodology with respect to the DTW metric achieves an improvement over linear resampling between 19% and 28% in the same tabulated cases.

PRD is certainly an important metric to test the correctness of the results, but, being the magnitude of the normalized distance vector, it carries little information about the overall morphology and position of the signal characteristics. Although the improvement in these regards is not significant, they are still useful in showing that, on average, the samples obtained with our method have similar distributions to the samples generated through standard resampling techniques (although with a very different density function, see Fig. 8).

DTW, on the other hand, is a useful tool to assess the morphological similarity of two vectors. We can observe how the DTW distance is already relatively small for linear resampled signal, and this is because, while missing all the details of the uniformly sampled signal, it still has a proportional evolution to its uniform counterpart: as the signal changes, thresholds are crossed, and the events drawn from the signal follow the same average behavior of the underlying uniformly sampled signal, making the linear resampling behaving similar in each segment but lacking all the details. Using our methodology, we are able to recover much of that fine details lost due to linear resampling, bringing a clear improvement to this metric.

Finally, delineation requires the precise location of P and T waves. While linear interpolation has a similar morphology to the uniformly sampled heartbeat, its peak and valleys (hence the characteristic defining apexes and ends of P and T waves) are bounded to be positioned at sampling instances. But this is not true in our methodology: between two events our method can fit any appropriate template section, without requiring the section to contain any maxima or minima. The same discussion can be made for inflexion points.

Hence, the recorded increase in DTW performance (i.e.: lower DTW distance between ground truth and reconstructed heartbeat) results in better positioning of P and T waves, while also reconstructing them better and with details more similar to the uniformly sampled signal, making the delineation more effective.

5.6. Different templates acquisition mode

As described in Section 3, we compare the performance of our processing pipeline with two computationally lighter, open loop, variations of it: 1) the templates set is computed only at the beginning of the process, 2) only one template is acquired at the beginning of the process.

Fig. 12 shows the distributions of the DTW distance for the three proposed processing pipeline variations, using an LC-ADC with either 3, 4, or 5 bits of dynamic range. These results highlight

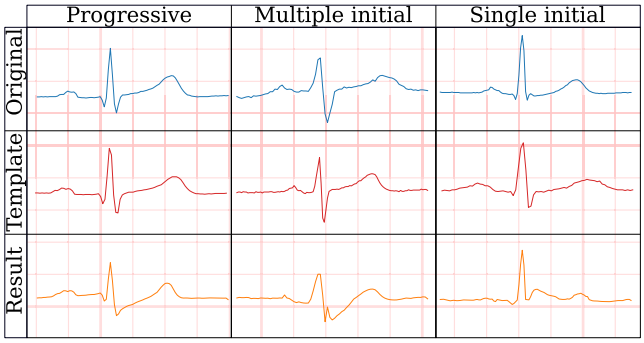

Fig. 13. Heartbeat reconstruction associated with the 50*th* DTW distance percentile, with an LC-ADC using 4 bits of dynamic range. Each column is a different templates-computation method. First column: progressive multi-template, second column: multiple templates acquired only at the beginning, third column: single initial template acquired at the beginning.

Table 6
Average and standard deviation PRD for linear interpolation, our processing pipeline using II-DDTW, and the same pipeline using standard DTW, with increasing EB-ADC bits.

ADC bits number	3	4	5
Linear interpolation	71.3 ± 27.1	56.8 ± 16.1	37.9 ± 14.3
Template based (standard DTW)	77.4 ± 27.0	58.5 ± 18.7	41.2 ± 17.0
Template based (II-DDTW)	70.5 ± 32.0	52.7 ± 19.8	36.0 ± 14.2

a condition of indifference under this type of measurement. Moreover, an analysis of the previously explored merit figures (PRD and F_1 delineation score) also shows no statistically significant differences between the three reconstruction methodologies. However, as shown in Fig. 13, the observation of the heartbeats in the 50th percentile of the DTW distance distributions shows a progressive worsening in the reconstruction as we use leMoreoverss adaptive and lighter reconstruction techniques.

Since every heartbeat of a patient shares common features [11], even only one generic heartbeat can be representative of a significant portion of a complete record. This makes the warping and the consequential main waves positioning as effective as using more specialized templates. However, as we show in Fig. 13, the lack of an adaptive technique to determine the best template for a set of events, and the absence of a mechanism to detect templates set significance, makes the solution not effective in representing the correct type of waves, especially in long-lasting recordings like the one obtained by a Holter ECG.

5.7. Standard DTW comparison

The final result to consider is the impact of the II-DDTW algorithm in our processing pipeline. In particular, we observe the difference in performances between the standard DTW algorithm and our implementation of the newly described II-DDTW. For all three metrics analyzed, we compare the performance measurements obtained for linear interpolation (the best-performing classical interpolation method), the processing pipeline as described in Section 3, and the same processing pipeline where II-DDTW has been replaced with the standard DTW algorithm.

First, we analyze the PRD results: Table 6 shows the resulting average and standard deviation of the PRD score in the three described cases. The approach using the standard DTW algorithm is slightly outperformed by II-DTW, and behave similarly, if not slightly worse, to linear resampling. As discussed previously in Section 5.5, this is mostly an indicator of the overall correctness of the samples distribution. Hence, while being numerically worse than the II-DDTW case, this measurement still carries the same information: the average re-sampled points distribution is similar, in average and standard deviation, between all three analyzed methods.

Table 7
Average and standard deviation DTW distance for linear interpolation, our processing pipeline using II-DDTW, and the same pipeline using standard DTW, with increasing EB-ADC bits.

ADC bits number	3	4	5
Linear interpolation	2.94 ± 1.70	2.42 ± 1.14	1.62 ± 0.71
Template based (standard DTW)	2.88 ± 1.78	2.38 ± 1.33	1.56 ± 0.76
Template based(II-DDTW)	2.16 ± 1.69	1.74 ± 1.19	1.32 ± 0.78

Table 8
P and T wave delineation results F_1 score for different resampling/reconstruction techniques and increasing EB-ADC bits.

ADC bits number	3	4	5
Linear interpolation - P	0.007	0.039	0.114
Linear interpolation - T	0.231	0.353	0.617
Template based (standard DTW) - P	0.438	0.634	0.626
Template based (standard DTW) - T	0.316	0.456	0.706
Template based (II-DDTW) - P	0.646	0.696	0.699
Template based (II-DDTW) - T	0.814	0.852	0.870

Table 7 presents the resulting average and standard deviation of the DTW distance in the three described cases. The approach using the standard DTW algorithm performs very similarly to the linear resampling, being outerformed by our implementation using II-DDTW. The lack in prior injection makes the templates section definition more challenging, resulting in a sub-optimal solution.

As shown in Table 8, using the standard DTW algorithm, the reconstruction delineation still outperforms the delineation of the linearly resampled signal. However, especially when using a low number of bits in our EB-ADC, the II-DTW algorithm still results in significantly better performances: when using standard DTW, P-wave detection worsens from 9% to 32% in the tabulated cases. For T-wave detection, the delineation results worsen from 20% to 61% in the same experiments. This performance worsening (in P wave, and especially in T-wave) is due to the fact that our approach (using II-DDTW) is especially effective when reconstructing signals with a very small number of samples (hence, low number of levels), while the improvement is more marginal (but still relevant) when more data is available. This behaviour is to be expected since the information prior we inject (the sampling timing) is less and less relevant for correct feature positioning, as more information about the signal is available.

Summarizing the discussed results, the usage of II-DTW instead of DTW lead to an average lower PRD and DTW distance, resulting in a better delineation of signals reconstructed when using our proposed methodology.

6. Conclusions

In this work, we have developed and implemented a signal processing pipeline able to reconstruct event-based (EB) sampled electrocardiogram (ECG) signals, through patient-specific heartbeat templates. This is accomplished by computing a set of locally representative heartbeats, selecting the best fitting one for each event-based sampled heartbeat, and warping it accordingly to the recorded events. We warp the templates using a novel formulation of the DDTW algorithm, named Information Injected-DDTW (II-DDTW), which uses the timing information of each event to bias the distance metric. The templates are dynamically re-computed whenever they stop being representative of the underlying physical signal, allowing the processing pipeline to select the most repre-

sentative template for each patient and signal section. When compared to standard resampling techniques (i.e., spline, sample-and-hold, and linear interpolation), we have shown that our proposed pipeline obtains a 10x improvement in P-wave reconstruction, a 2x improvement in T-wave reconstruction, and a 30% improvement, on average, in morphological similarity with the underlying physical signal.

Declaration of Competing Interest

Authors declare that they have no conflicts of interest

Acknowledgments

This work has been supported in part by the Swiss NSF ML-Edge Project (GA no. 200020_182009), the EC H2020 DIGIPREDICT Project (GA no. 101017915), and the EC H2020 DeepHealth Project (GA no. 825111). T.T. is supported by the grant RYC2021-032853-I funded by MCIN/AEI/ 10.13039/501100011033 and by the European Union NextGenerationEU/PRTR.

References

[1] World Health Organization, W.H.O. data on cardiovascular diseases, URL :https://www.who.int/health-topics/cardiovascular-diseases.
[2] L. Wei, et al., 13.3 A 7MB STT-MRAM in 22FFL FinFET technology with 4ns read sensing time at 0.9V using write-verify-write scheme and offset-cancellation sensing technique, in: 2019 IEEE International Solid- State Circuits Conference - (ISSCC), 2019, pp. 214–216, doi:10.1109/ISSCC.2019.8662444.
[3] S. Abadal, C. Han, J.M. Jornet, Wave propagation and channel modeling in chip-scale wireless communications: a survey from millimeter-wave to terahertz and optics, IEEE Access 8 (2020) 278–293, doi:10.1109/ACCESS.2019.2961849.
[4] I. Miro-Panades, et al., SamurAI: A 1.7MOPS-36GOPS Adaptive Versatile IoT Node with 15,000x Peak-to-Idle Power Reduction, 207ns Wake-Up Time and 1.3TOPS/W ML Efficiency, in: 2020 IEEE Symposium on VLSI Circuits, 2020, pp. 1–2, doi:10.1109/VLSICircuits18222.2020.9163000.
[5] A. Pullini, D. Rossi, I. Loi, A. Di Mauro, L. Benini, Mr. Wolf: A 1 GFLOP/s Energy-Proportional Parallel Ultra Low Power SoC for IOT Edge Processing, in: ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC), 2018, pp. 274–277, doi:10.1109/ESSCIRC.2018.8494247.
[6] F. Rincón, J. Recas, N. Khaled, D. Atienza, Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes, IEEE TITB 15 (6) (2011) 854–863.
[7] C.E. Shannon, Communication in the presence of noise, Proc. IRE 37 (1) (1949) 10–21, doi:10.1109/jrproc.1949.232969.
[8] G. Rovere, S. Fateh, L. Benini, A 2.1 μw event-driven wake-up circuit based on a level-crossing ADC for pattern recognition in healthcare, in: 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2017, pp. 1–4, doi:10.1109/BIOCAS.2017.8325145.
[9] S. Zanoli, F. Ponzina, T. Teijeiro, A. Levisse, D. Atienza, An error-based approximation sensing circuit for event-triggered, low power wearable sensors, arXiv:2106.13545
[10] D. Castro, P. Félix, J. Presedo, A method for context-based adaptive QRS clustering in real time, IEEE J Biomed Health Inform. 19 (5) (2015) 1660–1671, doi:10.1109/JBHI.2014.2361659.
[11] L. Sörnmo, P. Laguna, Chapter 6 - the electrocardiogram: a brief background, in: Bioelectrical Signal Processing in Cardiac and Neurological Applications, Academic Press, 2005, pp. 411–452, doi:10.1016/B978-012437552-9/50006-4.
[12] T. Teijeiro, P. Félix, J. Presedo, A noise robust QRS delineation method based on path simplification, in: 2015 Computing in Cardiology Conference (CinC), 2015, pp. 209–212, doi:10.1109/CIC.2015.7408623.
[13] A. Ibaida, D. Al-Shammary, I. Khalil, Cloud enabled fractal based ECG compression in wireless body sensor networks, Future Gener. Comput. Syst. 35 (2014) 91–101, doi:10.1016/j.future.2013.12.025. Special Section: Integration of Cloud Computing and Body Sensor Networks; Guest Editors: Giancarlo Fortino and Mukaddim Pathan
[14] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. 26 (1) (1978) 43–49, doi:10.1109/TASSP.1978.1163055.
[15] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, 2001, 10.13026/C2F305
[16] A.L. Goldberger, Z.D. Goldberger, A. Shvilkin, Clinical electrocardiography: a simplified approach,expert consult: online and print,8: clinical electrocardiography: A Simplified approach, ClinicalKey 2012, Elsevier/Saunders, 2012.
[17] B.B. Mandelbrot, Self-affine fractals and fractal dimension, Phys. Scr. 32 (4) (1985) 257–260, doi:10.1088/0031-8949/32/4/001.
[18] A. Aldroubi, K. Gröchenig, Nonuniform sampling and reconstruction in shift-invariant spaces, SIAM Rev. 43 (4) (2001) 585–620, doi:10.1137/S0036144501386986.
[19] R. Venkataramani, Y. Bresler, Optimal sub-Nyquist nonuniform sampling and reconstruction for multiband signals, IEEE Trans. Signal Process. 49 (10) (2001) 2301–2313, doi:10.1109/78.950786.
[20] N. Sayiner, H.V. Sorensen, T.R. Viswanathan, A level-crossing sampling scheme for a/d conversion, IEEE Trans. Circuits Syst. II Analog Digit. Signal Process. 43 (4) (1996) 335–339, doi:10.1109/82.488288.
[21] M. Taghouti, Chapter 10 - compressed sensing, in: F.H.P. Fitzek, F. Granelli, P. Seeling (Eds.), Computing in Communication Networks, Academic Press, 2020, pp. 197–215, doi:10.1016/B978-0-12-820488-7.00023-2.
[22] E.J. Keogh, M.J. Pazzani, Derivative dynamic time warping, pp. 1–11. 10.1137/1.9781611972719.1
[23] T. Giorgino, Computing and visualizing dynamic time warping alignments in R: the DTW package, J. Stat. Softw. 31 (7) (2009) 1–24, doi:10.18637/jss.v031.i07.
[24] Y. Jiang, et al., EventDTW: an improved dynamic time warping algorithm for aligning biomedical signals of nonuniform sampling frequencies, Sensors 20 (9) (2020), doi:10.3390/s20092700.
[25] I.M. Pu, Chapter 1 - introduction, in: I.M. Pu (Ed.), Fundamental Data Compression, Butterworth-Heinemann, Oxford, 2006, pp. 1–17, doi:10.1016/B978-075066310-6/50004-0.
[26] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, L. Sornmo, Clustering ECG complexes using hermite functions and self-organizing maps, IEEE TBME 47 (7) (2000) 838–848, doi:10.1109/10.846677.
[27] E.P. Hendryx, B.M. Rivière, D.C. Sorensen, C.G. Rusin, Finding representative electrocardiogram beat morphologies with CUR, J. Biomed. Inform. 77 (2018) 97–110, doi:10.1016/j.jbi.2017.12.003.
[28] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64, doi:10.1016/j.cosrev.2007.05.001.
[29] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976, doi:10.1126/science.1136800.
[30] M. Lepot, J.-B. Aubin, F.H.L.R. Clemens, Interpolation in time series: an introductive overview of existing methods, their performance criteria and uncertainty assessment, Water (Basel) 9 (10) (2017), doi:10.3390/w9100796.
[31] S. Zanoli, T. Teijeiro, F. Montagna, D. Atienza, An event-based system for low-power ECG QRS complex detection, in: 2020 Design, Automation Test in Europe Conference Exhibition (DATE), 2020, pp. 258–263, doi:10.23919/DATE48585.2020.9116498.
[32] A. Goldberger, et al. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals, 2000, 10.13026/C2F30
[33] T.W. Anderson, D.A. Darling, A test of goodness of fit, J. Am. Stat. Assoc. 49 (1954) 765–769, doi:10.2307/2281537.
[34] A.B. de Luna, V.N. Batchvarov, M. Malik, 1 the morphology of the electrocardiogram, 2005.
[35] T. Hofmann, B. Schölkopf, A.J. Smola, Kernel methods in machine learning, Ann. Stat. 36 (3) (2008) 1171–1220, doi:10.1214/009053607000000677.
[36] bluetooth core specification 5-3, URL: https://www.bluetooth.com/specifications/specs/core-specification-5-3/.
[37] S.M.S. Jalaleddine, C.G. Hutchens, R.D. Strattan, W.A. Coberly, Ecg data compression techniques-a unified approach, IEEE TBME 37 (4) (1990) 329–343, doi:10.1109/10.52340.
[38] A. Gray, Modern differential geometry of curves and surfaces with mathematica, Studies in advanced mathematics, 3rd ed./alfred gray, elsa abbena, and simon salamon, Chapman & Hall CRC, Boca Raton, FL, 2006, doi:10.1201/9781315276038.
[39] B. Vrcej, P.P. Vaidyanathan, Efficient implementation of all-digital interpolation, IEEE Trans. Image Process. 10 (11) (2001) 1639–1646, doi:10.1109/83.967392.