Toward Lightweight Biometric Signal Processing for Wearable Devices

Roberto Francescon , Mohsen Hooshmand , Matteo Gadaleta ,
Enrico Grisan , Seung Keun Yoon , Michele Rossi

***Abstract*— Wearable devices are becoming a natural and economic means to gather biometric data from end users. The massive amount of information that they will provide, unimaginable until a few years ago, owns an immense potential for applications such as continuous monitoring for personalized healthcare and use within fitness applications. Wearables are however heavily constrained in terms of amount of memory, transmission capability and energy reserve. This calls for dedicated, lightweight but still effective algorithms for data management. This paper is centered around lossy data compression techniques, whose aim is to minimize the amount of information that is to be stored on their onboard memory and subsequently transmitted over wireless interfaces. Specifically, we analyze selected compression techniques for biometric signals, quantifying their complexity (energy consumption) and compression performance. Hence, we propose a new class of codebook-based (CB) compression algorithms, designed to be energy efficient, online and amenable to any type of signal exhibiting recurrent patterns. Finally, the performance of the selected and the new algorithm is assessed, underlining the advantages offered by CB schemes in terms of memory savings and classification algorithms.**

I. Introduction

In the last decade, a large number of algorithms for the compression of electrocardiogram (ECG) signals has been proposed [1]–[5]. However, much less attention was paid to other biometric signals, which are becoming very common nowadays thanks to the advent of modern wearable devices. These, are spurring the development of new monitoring applications by, at the same time, providing prompt access to a number of vitals such as heart rate (from, e.g., photoplethysmographs), body temperature, ultraviolet radiation and movement data (gyroscope, accelerometer, etc.). These new technological developments are giving rise to the field of *human centric sensing* [6].

We emphasize that wearable technology is energy and memory constrained and, as such, efficient algorithms for the compression of the vitals are a must. Considering an ECG sampled at a typical rate of 250 samples per second with 12 bits per sample (e.g., from a Zephyr's Bioharness device) leads to 32.4Mbytes of data for a full day. Our compression algorithm can reduce this number by 80 times (405kbytes), leading to much higher efficiencies in terms of memory and transmission energy, as we show shortly below.

In this paper, we focus on on lossy compression for the sensed biometric signals. In Section II, we first classify previous algorithms. Thus, using *motif extraction* techniques [7], we design a novel online compressor for biometric data exhibiting recurrent patterns, see Section III. Finally, in Section IV the performance of selected and the new algorithm is assessed in terms of signal reconstruction fidelity and energy consumption. We underline that the computational complexity was often neglected in past studies, but this metric is of foremost importance for wearables. In Section V we conclude the paper, outlining avenues for future research.

Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35100 Padova, Italy. Samsung Advanced Institute of Technology (SAIT), 130, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, 443-803, Korea.

II. Common Lossy Compression Approaches

Existing compression algorithms for biometric signals can be grouped into three main families:

- **Transform based methods:** these exploit transformations such as Fast Fourier Transform (FFT) [3], Discrete Cosine Transform (DCT) [4] and Discrete Wavelet Transform (DWT) [5]. The rationale behind them is to represent the signal in a suitable transform domain and select a number of transform coefficients to be sent in place of the original samples. The amount of compression depends on the number of coefficients that are selected. Although the schemes belonging to this class have great compression capabilities, their computational complexity is often too high for wearable devices and, as we show shortly, higher than that of other solutions.
- **Time domain processing:** within this class we have AZTEC [1], CORTES [2] and Lightweight Temporal Compression (LTC) [8] [9]. AZTEC and CORTES achieve compression by discarding some of the signal samples and applying linear approximation, whereas LTC approximates the original time series through piecewise linear segments. These schemes are known for their lightweight character, but their compression and reconstruction capabilities are often regarded as inferior to those of transform based methods.
- **Parametric methods** these schemes use Neural Networks [10], Vector Quantization and Pattern Matching [11]. Their rationale is to process the temporal series to obtain some kind of knowledge and use it to predict the signal behavior. This is a field with limited investigation up to now. Also, these algorithms have promising capabilities for the extraction of signal features.

The new technique that we present in Section III belongs to the last category. It identifies recurrent patterns and builds a codebook (i.e., a *dictionary*) based on the most representative among them. Unlike previous solutions, our algorithm is not tailored to a specific type of signal (e.g., ECG) but can be applied to any biometric time series.

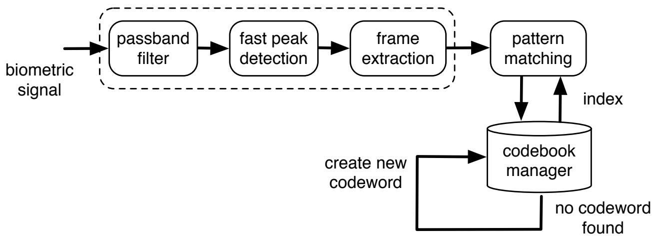

Fig. 1. Diagram of the codebook-based compression algorithm.

III. Codebook-based Lossy Compression

In this section we describe an original lightweight signal compression algorithm based on the concept of *motif extraction* [7]. Its main building blocks are shown in Fig. 1 and explained in what follows. The algorithm works with input biometric signals exhibiting recurrent patterns such as ECG, photo-plethysmographic traces (PPG), arterial blood pressure (ABP), respiratory signals (RS), etc. Our rationale is that recurrent patterns can be efficiently identified and used to build, *at runtime*, a codebook. This codebook has to be synchronized with the decompressor at the receiver and compression is achieved by sending, for each input pattern, the corresponding index in the codebook, in place of the original data sequence. We achieved this using a number of processing functions (see Fig. 1), including: 1) a passband filter, 2) a peak detector, 3) a frame extractor, 4) a codebook manager.

Passband filtering: as a first step, we apply a passband filter to remove artifacts. This filter currently operates in the band $8, 20$ Hz, although this can be easily tuned. Here, we implemented the third-order Butterworth filter of [12]. This processing is utilized to remove high frequency noise and the DC component.

Peak detection: here we have adopted the technique of [13] which has been conceived and tested especially for ECG signals. This algorithm is self-tuned and optimizes itself based on the input data sampling rate. We picked this scheme as it is very fast and lightweight and thus suitable for use in wearable and energy constrained devices. Also, although the algorithm has been tested with ECG traces, through minimal modifications, we also found it effective for PPG, ABP and RS signals.

Frame extractor: once the peaks are detected, we isolate the data samples between subsequent peaks. These constitute the input segments for our compressor algorithm. Note that, unlike the common practice of centering the frames around the peaks, we define a segment as the data sequence *between subsequent peaks*. This allows using machine learning algorithms for the construction of the codebook, as we detail shortly.

Codebook manager: this block has a pivotal role in our signal compression framework. It is loosely based on *vector quantization* [14] and has two main functions: 1) *maintaining* a consistent and well representative codebook, 2) *encoding* input patterns into the corresponding indices from the codebook. Let $\boldsymbol{x}_t$ be the segment provided by the pattern matching block at the generic time t 0 (time is discrete). Also, with $\mathcal{C}_t$ $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ we indicate the codebook at time t and with c_i, i $1, \ldots, N$, the corresponding codewords. Segment $\boldsymbol{x}_t$ is remapped into a new segment $\boldsymbol{y}_t$ of predefined length (W samples), where $\texttt{size}\ \boldsymbol{c}_i$ $\texttt{size}\ \boldsymbol{y}_t$ W, for i $1, \ldots, N$. The new segment $\boldsymbol{y}_t$ is obtained using a classical resampling procedure and removing offset o_t and gain g_t from $\boldsymbol{x}_t$ (see equations (5)–(7) of [7]).[1] Thus, a suitable distance $d\ \boldsymbol{y}_t, \boldsymbol{c}_i$ is evaluated for all words $\boldsymbol{c}_i$ in the codebook and the one with the minimum distance, with index i , is selected. Now, if $d\ \boldsymbol{y}_t, \boldsymbol{c}_i$ ε, codeword i is deemed a good representative for the current segment, otherwise $\boldsymbol{y}_t$ is added to the codebook as a new codeword, where with i we mean the associated index. In the last case, the new codeword has to be communicated to the compressor. ε is a tunable parameter that we use to control the signal reconstruction fidelity at the decompressor. Finally, the index i is sent in place of the full segment, along with o_t, g_t and its original segment length, ℓ_t.

We remark that several distance functions can be used in the manager, the L -norm has been considered for the results in this paper as we found it performed satisfactorily across a large range of signals. To approximate a segment, the decompressor applies three transforms to segment i from the codebook: renormalization with respect to offset o_t and gain g_t and resampling according to the actual segment length ℓ_t. The removal of codewords from the codebook can be implemented based on *last used* timestamps, although this aspect is not evaluated here and is left as a future work.

IV. Performance Evaluation

Next, we show quantitative results for the considered signal compression algorithms, detailing their energy consumption, compression and reconstruction fidelity performance. The *energy consumption* has been computed for all compression algorithms by taking into account the number of operations performed by the micro-controller unit (MCU), i.e., the number of summations, multiplications, divisions and comparisons. These have been subsequently translated into the corresponding number of cycles and, in turn, into the energy consumption in Joule/bit considering the Cortex M4 as a reference processor. The *compression efficiency* has been computed as the ratio between the total number of bits required to transmit the original signal divided by those required for the transmission of the compressed one. For the *reconstruction fidelity*, we computed the root mean square error (RMSE) between the original and the compressed signals. Next, we show results for ECG and PPG considering biometric traces from the MIT-BIH and Physionet MIMIC II databases [15].

As a first set of results, in Fig. 2 we plot the RMSE as a function of the compression efficiency for the considered algorithms. Here, the RMSE has been expressed as a percentage of the average peak-to-peak amplitude of the input signal. Two variants of DCT and DWT have been implemented.

[1] A linear resampling technique is used for the results in this paper, although other schemes based on quadratic and Spline interpolation can also be used.

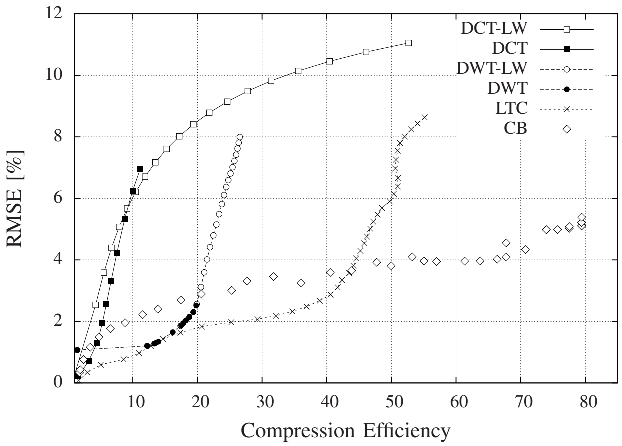

Fig. 2. ECG signal: RMSE *vs* Compression efficiency.

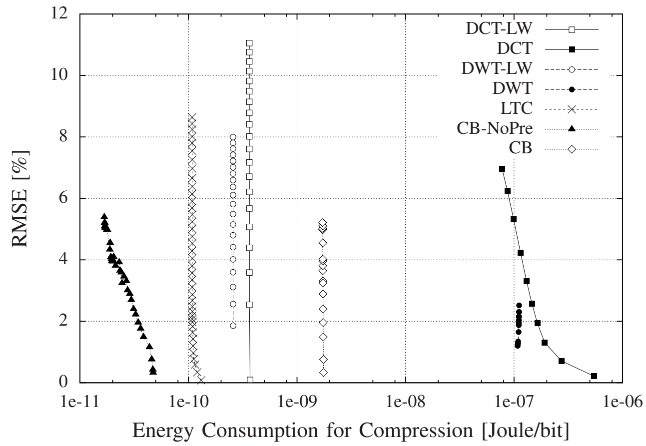

Fig. 3. ECG signal: RMSE *vs* energy consumption for compression.

In the first implementation (labeled DCT and DWT), the compressor operates according to a controlled *maximum error tolerance*, provided as an input parameter. Thus, at each iteration, it checks how many and which transform coefficients are to be retained (and sent) for maximum compression performance, while meeting the given error requirement (see [16]). A second and lightweight implementation (DCT-LW and DWT-LW) considers a threshold on the amount of signal energy that is captured by the selected coefficients, irrespective of the reconstruction error at the decompressor. Note that LTC and our own codebook-based (CB) scheme of Section III only operate according to the first approach, i.e., setting the error tolerance. From Fig. 2, we observe that there is no single best algorithm: some schemes perform best in terms of RMSE and others in terms of compression efficiency and they also differ in terms of maximum achievable compression. At low compression efficiencies, DCT and DWT provide the best reconstruction fidelity. Their lightweight variants perform worse because they have no way to assess which coefficients are the best to be retained in terms of RMSE. As already observed in [16], LTC has excellent results in terms of RMSE, however its linear approximations only preserves the R peaks (in the ECG case), while losing details elsewhere. CB performs satisfactorily at low compression ratios and, most importantly, maintains the RMSE quite small (see its trend for an increasing compression efficiency) showing the highest compression ratios, i.e., as high as 80 times for the considered signal and 30% greater than the maximum compression efficiency of LTC.

In Fig. 3, we show the RMSE against the energy drained by the compression algorithms. As expected, DCT and DWT are the most energy demanding and this is due to their error checks in the selection of the coefficients, which imply the execution of inverse transforms at the compressor. LTC is energy efficient and is in fact a good candidate, despite its simplicity (linear approximations). CB performs in the middle between DCT / DWT and LTC. However, we underline that the energy consumption of CB is dominated by the bandpass filter and the peak-detection blocks of Fig. 1. To see this, we have plotted the performance of CB without the pre-processing blocks (termed "CB-NoPre"). As anticipated, CB-NoPre is the most energy efficient algorithm. Note that filtering is always performed to remove measurement artifacts and peak detection is also very often utilized to extract relevant signal features such as the *instantaneous heart rate*. Given this, the energy consumption associated with the required pre-processing functions may not be a problem, especially if these functions are to be executed anyway. In Fig. 4, we show the compression efficiency against the *total* energy consumption, where the latter is obtained summing the energy required for compression and the subsequent transmission of the compressed signal over a Bluetooth LE wireless interface (a Texas Instruments CC2541 radio was considered for the results in this paper). From this plot, we see that the total energy consumption is dominated by the transmission energy. As a consequence, the compression efficiency is key to reducing the energy drained, but different algorithms do not show noticeable differences.

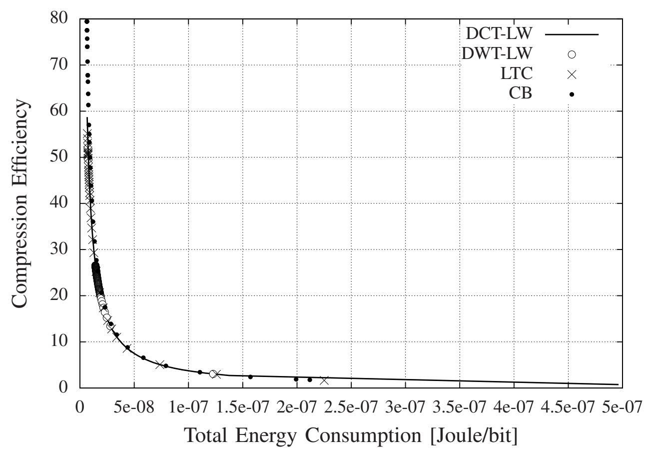

Fig. 4. ECG signal: compression efficiency *vs* total energy consumption.

The codebook size is quantified in Fig. 5. From this graph, we postulate that it may make sense to use CB for compression efficiencies, e.g., greater than 40. Within this range, the number of patterns in the dictionary is smaller

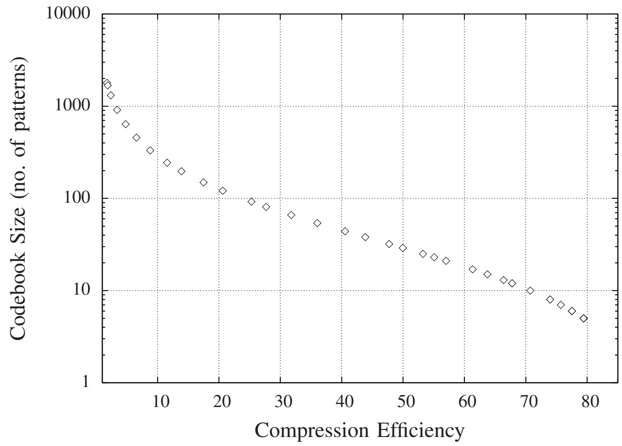

Fig. 5. ECG signal: codebook size N *vs* compression efficiency.

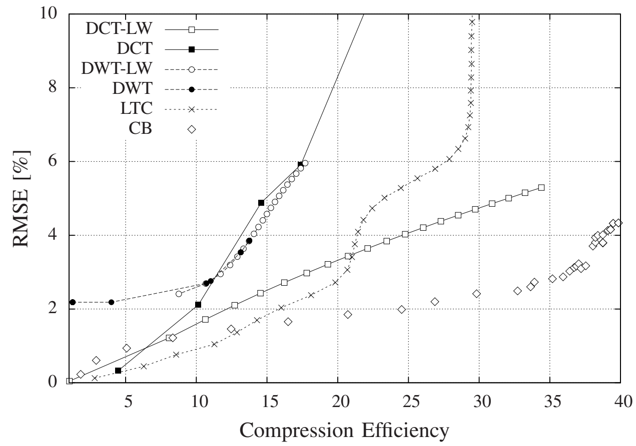

Fig. 6. PPG signal: RMSE *vs* compression efficiency.

than 35, while the RMSE remains acceptable (i.e., within the range $4,6$ % of the signal's peak-to-peak amplitude, see Fig. 3), surpassing all the other schemes.

In the last Fig. 6, we show the RMSE *vs* compression efficiency performance for a typical PPG trace. These results confirm those in Fig. 2 and the same conclusions where also obtained for the other metrics. Note that for PPG the maximum compression efficiency is 40 and this depends on the lower signal's sampling rate. We emphasize that the described index-based compression strategy has a theoretical limit on the maximum achievable compression efficiency, which depends on the number of samples per segment for the original signal versus the minimum number of features that are transmitted by the compressor, namely, the index i , o_t, g_t and the segment length ℓ_t. We stress that this theoretical limit respectively corresponds to 80 and 40 times for ECG and PPG, which are both reached by CB. Similar trends, not shown due to space constraints, where also found for ABP and RS traces.

V. Discussion and Future Work

In this paper, we have proposed a novel approach to the design of codebook-based compression algorithms based on motif extraction and pattern matching. This approach was found to be superior to existing schemes in terms of compression efficiencies, whilst retaining good reconstruction accuracies at the decompressor. We remark that, in addition to its compression capabilities, codebook-based schemes enable further processing functions such as classification and learning. In fact, the codewords (patterns) in the dictionary and the related number of uses per codeword can be utilized to assess statistical properties of the corresponding signals, including their most relevant features. Our future developments are centered on these facts and on the study of self-adapting dictionaries based on neural networks. These, in addition to learning codewords at runtime, as we do here, will also be capable of adapting them according to changes in the signal statistics. Finally, we are exploring means to extend the proposed pattern matching approach to signals from other sources, such as wearable 9-axis motion tracking sensor devices.

References

[1] J. Cox, H. Fozzard, F. M. Nolle, and G. Oliver, "AZTEC: A preprocessing system for real-time ECG rhythm analysis," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 9, pp. 128–129, 1968.

[2] J. P. Abenstein and W. J. Tompkins, "A New Data-Reduction Algorithm for Real-Time ECG Analysis," *IEEE Transactions on Biomedical Engineering*, vol. 29, no. 1, pp. 43–48, 1982.

[3] R. Shankara and S. M. Ivaturi, "ECG Data Compression Using Fourier Descriptors," *IEEE Transactions on Biomedical Engineering*, vol. 33, no. 4, pp. 428–434, 1986.

[4] H. Lee and K. M. Buckley, "ECG data compression using cut and align beats approach and 2-D transforms," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 5, pp. 556–564, 1999.

[5] B. A. Rajoub, "An Efficient Coding Algorithm for the Compression of ECG Signals Using the Wavelet Transform," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 4, pp. 355–362, 2002.

[6] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric Sensing," *Phylosophical Transactions of Royal Society*, vol. 370, no. 1958, pp. 176–197, 2012.

[7] M. Danieletto, N. Bui, and M. Zorzi, "RAZOR: A Compression and Classification Solution for the Internet of Things," *Sensors*, vol. 14, no. 1, pp. 68–94, 2014.

[8] T. Schoellhammer, B. Greenstein, M. W. E. Osterweil, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in *IEEE International Conference on Local Computer Networks (LCN)*, Tampa, FL, US, Nov. 2004.

[9] M. Ishijima, S. Shin, and G. Hostetter, "Scan-along polygonal approximation for data compression of electrocardiograms," *IEEE Transactions on Biomedical Engineering*, vol. 30, no. 11, pp. 723–729, 1983.

[10] N. Maglaveras, T. Stampkopoulos, K. Diamantaras, C. Pappas, and M. Strintzis, "ECG pattern recognition and classification using nonlinear transformations and neural networks: A review," *International Journal of Medical Informatics*, vol. 52, no. 1–3, pp. 191–208, 1998.

[11] J. Cardenas-Barrera and J. Lorenzo-Ginori, "Mean-shape vector quantizer for ECG signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 1, pp. 62–70, 1999.

[12] N. M. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of First-Derivative Based QRS Detection Algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 478–484, 2008.

[13] M. Elgendi, "Fast QRS Detection with an Optimized Knowledge-Based Method: Evaluation on 11 Standard ECG Databases," *PLoS ONE*, vol. 8, no. 9, pp. 1–18, 2013.

[14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, US: Kluwer Academic Publishers, 1991.

[15] M. Saeed et al., "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database," *Critical Care Medicine*, vol. 39, no. 5, pp. 952–960, 2011.

[16] D. Zordan, B. Martinez, I. Villajosana, and M. Rossi, "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking," *ACM Transactions on Sensor Networks*, vol. 11, no. 1, pp. 15:1–15:34, Nov. 2014.