Università degli Studi di Padova

Design and Implementation of a Motif-based Compression Algorithm for Biometric Signals

Laurea Magistrale in Ingegneria delle Telecomunicazioni

Laureando:
Roberto Francescon

Relatore:
Prof. Michele Rossi
Correlatore:
Davide Zordan

Signet
Dipartimento di Ingegneria dell'Informazione

Anno Accademico 2014/2015

UNIVERSITÀ DEGLI STUDI DI PADOVA

Abstract

Scuola di Ingegneria
Dipartimento di Ingegneria dell'Informazione

Laurea Magistrale in Ingegneria delle Telecomunicazioni

Design and Implementation of a Motif-based Compression Algorithm for Biometric Signals

Roberto Francescon

Wearable devices are becoming a natural and economic means to gather biometric data from users. The massive amount of information that they provide, unimaginable until a few years ago, owns an immense potential for applications such as continous monitoring for personalized healthcare and use within the fitness field. Wearables are however heavily constrained in terms of memory size, transmission capabilities and energy reserve. This calls for dedicated, lightweigth yet effective algorithms for data management. This thesis is centered around lossy data compression techniques, whose aim is to minimize the amount of information that is to be stored on mobile devices and subsequently transmitted over wireless interfaces. Specifically, a number of selected compression techniques for biometric signals are analyzed and their complexity (energy consumption) and compression performance are quantified. Hence, a new class of codebook based (CB) compression algorithms is proposed, 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 algorithms is assessed, highlighting the advantages offered by the proposed CB schemes in terms of energy savings and classification properties.

Acknowledgements

Prima di tutti vorrei ringraziare Yasmin, "*di grandi sconvolgimenti origine piccina*" e, ovviamente, la mia famiglia, senza la quale, nella vita, non avrei iniziato nulla.

I miei sinceri ringraziamenti vanno anche a tutti i membri del laboratorio Signet (parte DEI/G), il cui supporto è stato fondamentale: Davide, Filippo, Matteo D., Matteo G., Mohsen, Riccardo e Waqas.

My sincere thanks also goes to all the Signet laboratory members (DEI/G part), whose support was vital: Davide, Filippo, Matteo D., Matteo G., Mohsen, Riccardo and Waqas.

Contents

List of Figures

List of Tables

Abbreviations

BAN	**B**ody **A**rea **N**etwork
CPU	**C**entral **P**rocessing **U**nit
DCT	**D**iscrete **C**osine **T**ransform
DFT	**D**iscrete **F**ourier **T**ransform
DSP	**D**igital **S**ignal **P**rocessing
DWT	**D**iscrete **W**avelet **T**ransform
ECG	**E**lectro**C**ardio**G**ram(**G**raphy)
FFT	**F**ast **F**ourier **T**ransform
FPU	**F**loating **P**oint **U**nit
IoT	**I**nternet **o**f **T**hings
LTC	**L**ightweight **T**emporal **C**ompression
LTE	**L**ong **T**erm **E**volution
MM	**M**athematical **M**orphology
NN	**N**eural **N**etworks
PPG	**P**hoto**P**lethysmo**G**ram(**G**rphy)
PLA	**P**iecewise **L**inear **A**pproximation
RMSE	**R**oot **M**ean **S**quare **E**rror
SOM	**S**elf **O**rganizing **M**aps
UMTS	**U**niversal **M**obile **T**elecommunications **S**ystem
VQ	**V**ector **Q**uantization
WiFi	**Wi**reless **Fi**delity
WSN	**W**ireless **S**ensor **N**etwork

To Satan, our light through the darkness of ignorance

Chapter 1

Scenario

1.1 General concept of IoT

In the last decade there has been the diffusion of a very large number of emerging technologies that enabled the gathering of data through a variety of sensors, operating in many different fields (e.g environmental, mobility-related, medical). The so collected data, as a result, gave rise to the development of applications able to organize it, analyze it a finally present it in a suitable way, thus allowing for substantial improvements in human activities at large.

This concept grew in what is now called the *Internet of Things* (IoT) paradigm: the scenario is that of a very large number of sensor devices with communication capabilities, connecting both in an ad-hoc fashion or to a sink node, which collects data for further elaboration. These sensor nodes would not have high capabilities in terms of computational power and self-sustainability. This framework, according to the IoT vision, will be realized by enhancing everyday objects with a communication interface and then expoliting their ability to sense and collect data.

Up to now, the development of the Internet of Things, has proceeded by proliferation of *islands*, i.e. groups of devices able to communicate and exchange information that are, due to multiple incompatibilty issues, unable to communicate with devices of other IoT islands. In [1], the authors examine this fundamental problem and suggest directions that are to be taken in order

to reach a *real* Internet of Things: enabling, not only the collection of the sensed data in the physical world, but also the exchange and elaboration of it in the digital domain.

This convergence should take start, again according to [1], from the design of a general architecture, able to integrate existing efforts and current technologies and thus overcoming the current fragmentation where "*many INTRAnets of things cannot operate in an well-integrated INTERnet of Things*". Thus, it is not advisable to design from scratch a communication architecture that would, probably, just increase the number of different and non-interoperable communication architectures, but rather to provide a sort of *middleware* that would let, already operating technologies, to be interoperable.

1.2 Wearable Devices

A category of devices whose presence has become prevasive in the very last years is that of *wearable devices*: if we include in this class also *smartphones*, then it is clear that an immense potential was unleashed as these devices, originally made for communication, nowadyas ship a large number of sensors. As noted in [2], the IoT paradigm has recently shifted from a scenario in which sensors are integrated in the environment to one in which we, as humans, carry ourselves the sensor devices and partecipate actively in the sensing process.

As stated in [2], it is observable that a conjunction in the spread of two technology advancements has boosted the development of applications for which the sensing process has a personal and social character: the first technology involves the update of the communication network and the second the diffusion of inexpensive sensing devices. The rapid improvoments which the cellular network has undergone are in front of everyone and they are paralleled by the update of the core network. The terminals too, namely the now-called *smartphones*, besides powerful processors, carry, nowadays many sensing devices which targets different fields: imaging (cameras), position (GPS), movement (accelerometer and gyroscope) and geomagnetical field (magnetometer) among others. Also, through a number of applications, e.g. Twitter, Facebook, Whatsapp, etc., it is possible to collect and share massive amounts of data.

New small and lightweight wearable devices for the sensing of biometric signals are also appearing in the market, thus pushing the interest of reseachers and posing new challenges. The first section of market to see a spread of adoption has been *fitness*: wristbands able to communicate to the mobile phones (usually via Bluetooth) started to integrate accelerometers that could give sufficiently accurate estimate of the number of steps walked, UV light sensors that can measure how much solar radiation one have been exposed to and, most importantly, photodiodes able to generate a *photoplethysmogram* (PPG) signal. A possible second section of market, which is, according to the author, the next one which will go through rapid expansion, is that of biomedical devices for the monitoring of non critical patients such as elderly people or patients who have to undergo Holter exam: this, of course, entails a shift in requirements from energy-economy to reliability.

1.2.1 Fitness

As already noticed, the fitness market was the first to see the appearence of products directly involved in the measurements of biometric quantities: most modern smartphones already implemeted some of these functions but dedicated devices let people not owning cutting-edge smartphones, to benefit of these technologies and to shape the device on the purpose. Dedicated fitness devices could be thus improved by adding more sensors to collect more quantities, and, most importantly, saving energy, given that they incorporate a separate battery.

The proliferation of these devices was spurred by factors similar to those that made smartphones so widespread: first of all the integration of reliable sensors that have become quite cheap and then the possibility to interface them with the internet, thus storing online, sharing and analyzing the signals they collect. Among others, the possibilty to generate PPG signals form *photodiodes* positioned on the wrist, as proven by some of the devices already in the market, has impressed a faster pace to their diffusion. The PPG signals generated this way led to the possibility of estimating an important parameter such as the *Heart Rate* (HR): the main issue that wearable devices have to overcome is motion noise, which, under intensive exercise could be very disrupting. This problem along with others, though not completely solved, is already analyzed and studied in works such as [3] [4].

A minor group of more sophisticated and advanced devices such as the BioHarness form Zephyr [5], are able to collect a proper ECG signal (single lead). These are devices that, even though available on the market, are targeted to professionals and athletes at high levels and their cost confirms this assertion.

1.2.2 Medical

The hospital wards have different requirements from devices which retrieve biometrical signals, although some are shared with the fitness world. A lasting battery would be appreciated also by this last category but precision and reliability are of foremost importance, given that we are talking about unhealthy people. Thus the possible sources of errors should be bounded or errors should be reported instantly in some way. In fact, in the Hospitals, biometric signals are gathered through machines that are often large, expensive and have very high computational capabilities (and are, implicitly, not portable); these devices were made with the purpose of being reliable and precise, given that they are mainly used in scenarios such as Intensive Care Units. Another important factor, which could seem unimportant at a first sight, is cost: these devices should be accessible to as much institutions as possible given that, in many Public Health Hospitals, their adoption will be directly influenced by the costs.

This work is targeted to health devices with the vision that, in the coming years, assistance to people affected by not serious patologies, could be automated through the use of portable and inexpensive devices to let, for example, these people continue living at their houses; another improvement could be done inside the Hospital wards where, very often, the staff is undersized. The challenge is, thus, to develop the fitness devices, already available, and to port them to a more technical level of application.

Chapter 2

Compression approaches

2.1 Compression in Sensor Networks

This work is focused on temporal compression of biosignals, with the aim of allowing battery-operated devices to save energy: it is known, indeed, that the most part of the energy expenditure of a mobile device is given by its wireless communication interface [6]. This fact led, in the past, to a common pitfall: the concern to compress as much as possible, very often obfuscated the fact that the processor too, when compressing, consumes energy; there exists thus the possibility for this energy to be more than that required for sending the data without any compression. A complete study of this possibility was accomplished in [6], where a number of lossy compression techniques, spanning a large number of different approaches, were tested in terms of compression capabilities and energy expenditure. The reconstruction capability of the algorithms was taken out of the study by implementing all of them in a way, that let the tolerance, i.e. the difference between original and reconstructed sequence, to be set as a parameter common to all. The results show that there is actually a threshold below which compression becomes inefficient, for it entails more energy than that needed for uncompressed data forwarding. It is remarked that, although compression can be *lossless* and *lossy*, this work only considers lossy techniques.

A group of appealing techniques are also advised in [6], as these techniques are beyond, or near to, the efficiency treshold: among these there is the *Discrete Cosine Transform* (DCT), for the transform based approach, and the *Lightweight Temporal Compression* (LTC), for the linear approximation approach (they will be explained in detail later). For this reason, these techniques are considered for comparison later in this work, even though in [6] a very constrained hardware architecture was considered, such as the MSP430, which is far a different choice from that made in this work.

2.2 Biometric signals

Biometric signals, or biosignals, are signals generated form the activity of the body during its everyday functions; these signals, collected as voltages or currents through some types of transducers, can have different nature: chemical, electrical, mechanical and magnetical and can be instrumental to the diagnosis process. Among the most commonly analyzed functions there are respiration, heart activity, brain activity and many others. A first dicotomy that can be made involves the presence of some repetition: signals such as ECG, PPG, Respiratory Signal and Arterial Blood Pressure (ABP) clearly present some sort of repeatability given by the cycling activity of the organs involved; other type of signals, on the contrary, do not exihibit a high degree of repetition.

In this work only repetitive signals will be taken into consideration, for which the redundancy can be removed thus achieving compression. The word *repetitive* was used, rather than periodic, to highlight the fact that the repeatability occurs with some variations. These variations occur both in time and amplitude and, although random, it can be seen that they have a strong correlation from a quasi-period to the next one. This is true, however, only for the *normal* behavior, that is, for a healthy individual: the challenge is thus to effectively remove the redundancy without losing important diagnostic information.

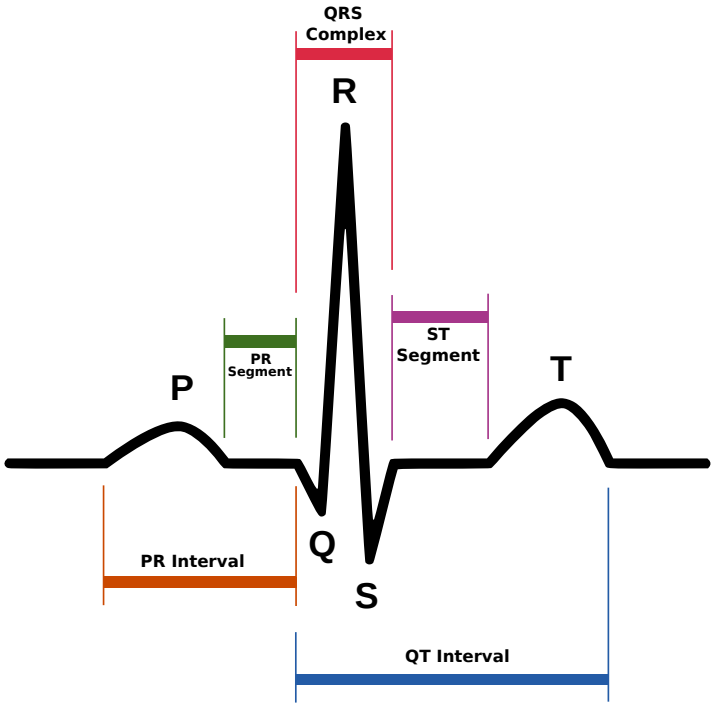


FIGURE 2.1: Basic structure of a typical QRS complex

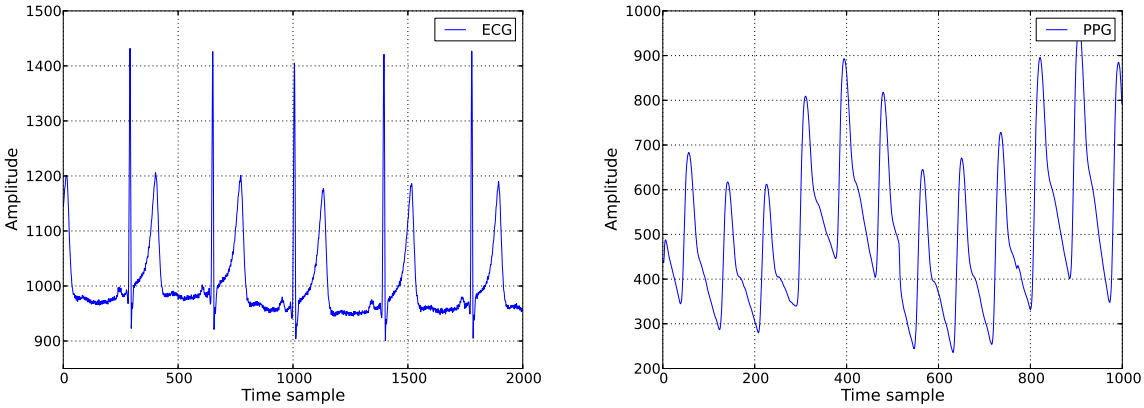
2.2.1 Examples

ECG The heart pumps blood through arteries and veins, permitting it to oxygenate organs and to carry out carbon-dioxide; to this aim, it follows a cycle in which, it generates voltage differences which activates its muscular structure and that are measurable through the skin immediately above it. The myocardial activity of the heart is induced by a signal which structure has been deeply studied: it is formed by three groups of components and totally by five components: these are the P-, Q-, R-, S- and T-wave, among which the Q-R-S complex forms a special group. A typical P-QRS-T period of an ECG trace from a healthy person is shown in Figure 2.1.

We note that the most high-frequency component is the so-called *R peak*: it is indeed this peak that is used to keep track of the Heart Rate (HR); however all the other components still convey important information about the health state of a peron's heart and are, thus, to be preserved in their shape.

As already mentioned, there exists a large amount of literature on efficient methods for locating the R peaks: among these the most widely used has certainly been the Pan-Tompkins algorithm [7]. Using a sequence of passband filtering, squaring, integrating and tresholding it proved to be a very effective procedure and it remained, until today, a very robust R peak detector. Lately, some works made large and exhaustive reviews on QRS complex detection algorithms, such as [8], while others already considered the scenario of wearable devices and made device-oriented reviews [9].

Machines able to collect the ECG signal were, in the past, only present in Hospitals and Healthcare structures, mainly in the form of *Intensive Care Unit* (ICU) bedside workstations. In the last decade a small number of new portable devices appeared in the market, which are able to collect a single lead ECG track such as the already mentioned Bioharness module [5].



(A) Typical ECG normal trace (B) Typical PPG normal trace

FIGURE 2.2: Sample traces for ECG and PPG

PPG Photoplethysmography (PPG) is a simple but effective technique used to measure volume changes in the microvascular bed of tissue: basically it works by measuring the light reflected, or refracted, by the capillaries. The PPG waveform, as can be seen in Figure 2.2b, comprises a pulsatile (AC) physiological waveform attributed to cardiac synchronous changes in the blood volume with each heart beat, and is superimposed on a slowly varying (DC) baseline with various lower frequency components. Photoplethysmography can also be used to asses oxygen saturation (SpO_2), blood pressure and cardiac output [10].

The pulse oximeter is a typical medical application of the photoplethysmography: it is a device used to monitorate, at the same time, blood oxygenation and heart rate. It is based on the physical principle according to which the colour of the blood depends on the amount of saturated hemoglobin (HbO_2). Saturated hemoglobin is red-coloured contrary to desaturated hemoglobin, which is blue-coloured. Hemoglobin, indeed, changes its structural configuration when it is involved in a chemical reaction; each configuration shows a different behavior in reflecting and refracting light: so, two different wavelengths are used to detect different saturation levels.

In recent years many fitness bands available on the market, started to integrate a photoplethysmograph able to collect PPG signals while positioned on the wrist: a LED paired with a photodiode are used to measure blood volume changes and, subsequently, to extract the Heart Rate (HR).

Other biosignals Other types of signals may include Arterial Blood Pressure (ABP) and Respiration Signal (RS). ABP is measured through a cuff, once put on the upper part of the arm and now also on the forearm, which once inflated, measures the pressure of the arteries by the *oscillometry* method. Respiration, on the other hand, must be derived, through some signal processing, from the previously described signals, namely ECG, PPG or both.

2.2.2 Common compression techniques

Most of the literature in the field of compression for biometric signals, is devoted to compression schemes to be applied to the ECG: little was written for what concern other types of biosignals. The compression algorithms for ECG signals that have been developed so far, can be subdivided into three main families:

- **Transform based methods**: these methods exploit tranformations such as the Discrete Fourier Transform (DFT) [11], Discrete Cosine Transform (DCT) [12] and Discrete Wavelet Transform (DWT) [13]. The rationale behind them is to represent the signal in a number of suitable transform domain and select a number of coefficients to send

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 will show later, higher than the complexity of other solutions.

- **Time domain processing**: within this class we have historical algorithms such as AZTEC [14], CORTES [15] and more recent ones such as Lightweight Temporal Compression (LTC) [16] [17]. AZTEC and CORTES achieve compression by discarding some of the signal samples and applying linear approximation, whereas LTC approximates the original sequence by calculating, sample by sample, the best segment. These schemes are known for their lighweight nature, but their compression and, especially, reconstruction capabilities are often regarded as inferior to those of transform based methods.

- **Parametric models**: these schemes use mainly Neural Networks (NN) [18], Vector Quantization (VQ) and Pattern matching [19]. Their rationale is to process the temporal series to obtain some kind of knowledge and use it ot predict the signal behavior. This is a field with limited investigation up to now, specially because it does not include recent advancements in the *data mining* and *pattern matching* fields. These algorithms, however, have promising capabilities in extracting the signal features.

The new technique that will be presented in this work belongs to the last category: it identifies recurrent patterns and builds a codebook (or dictionary) based on the most representative among them. Unlike previous solutions, our algorithm is not tailored to a specific type of biometric signal, and can be applied to any quasi-periodic time series.

2.2.3 Related work

Other works investigates the use of pattern recognition and Vector Quantization (VQ) for the purpose of compressing biometric signals, and in particular the ECG. The authors in [20] advise a method based on VQ, in which vectors, as in our case, are considered to be samples between successive peaks: it subdivides, based on the different length of the input vectors, the

dictionary in many subsets. In [21] VQ is applied, but to a portion of the heart beat centered in the R peak, although it is not clear how the method chooses the discarded samples.

The algorithms whose structure is most similar to ours, by taking as input vectors portions of the signal between peaks, exploit VQ in a way that does not lend itself to online application. In [22], as an example, the authors propose a clever way to compare time series having different length, but the dictionary is built using a classical Linde-Buzo-Gray offline procedure.

The schemes which are most similar to what is proposed in this work are found in [23] and [24]. The work in [23] has, in common with our proposed algorithm, only the use of a resampling process as available from *multirate digital signal processing* to normalize, in time, the heartbeats series and then to exploit usual norms as defined in standard multidimensional vector spaces. The work in [24] shares with our proposed method the capability of working in a online fashion. The authors propose an algorithm which exploits VQ using a novel type of dictionary: it works on a single row vector where patterns are put one after the other and their occurences determine the possibility, for them, to be added and even kicked off. The authors, however, do not explain how different length patterns, as are the heartbeats, are compared using the L_2-norm neither what they mean with *input vectors*: the paper proposes a method to overcome the online applicability problem but does not advise a scheme to embed variable length vectors into the general Vector Quantization framework.

Chapter 3

A Motif-based and Vector Quantization framework

3.1 Fundamentals

The framework at the core of the proposed algorithm originates from two fundamental techniques which have already proven their effectiveness and whose capabilities has been exploited in various fields: *Motifs extraction* and *Vector Quantization*.

3.1.1 The concept of Motif

The concept of *Motif* originated from the field of time series *data mining* and *pattern recognition* to which the group of Keogh [25][26] contributed in a substatial way. The term (having as synonyms expressions such as *recurrent pattern* or *primitive shapes*) was taken from *computational biology* and aims at defining subportions of a signal which can be of variable dimension and appear with a certain degree of frequency (which, however, is expected to be high). In this section the basic concepts are given, which are necessary to understand what comes next.

The problem of efficiently finding a given pattern in a database was, in last decades, studied and analyzed to a great extent and can be considered, since long ago, a solved one. On the

contrary, identifying the most recurrent patterns and their occurences in a given signal was not studied such deeply. A formal and complete definition of the problem is effectively given in [25], where an algorithm to discover the *Motifs* is proposed. The work in [25] focuses, as we need, on time sequences. Given a time series of length N:

$$\mathbf{X} = \{x_1, x_2, ..., x_n, ..., x_N\}, \tag{3.1}$$

a subsequence, of length M ($M < N$, but usually $M << N$), of the given sequence is defined as

$$\mathbf{x} = \{x_m, x_{m+1}, ..., x_{m+M-1}\},\ \forall m \leq N - M + 1. \tag{3.2}$$

It is now clear that we are looking for those subsequences which occur with the highest frequency. To define the *occurrence* we account for a small difference among sebsequences, that is, we define a function to formally express *dissimilarity*, $D(\mathbf{x}_i, \mathbf{x}_j)$ according to which we have a *match* if the so-defined distance does not exceed a treshold ε:

$$D(\mathbf{x}_i, \mathbf{x}_j) \leq \varepsilon \Rightarrow \mathbf{x}_i \text{ match } \mathbf{x}_j. \tag{3.3}$$

It is worth noting that this kind of *match* is a relation which enjoys the symmetric and reflexive properties, but not the transitive property.

Now, to discover the most frequent patterns, the authors of [25] define the *1-Motif*, $\mathbf{x}_1$, to be the pattern with the highest count of matches; the general *K-Motif*, $\forall K > 1$ is the pattern with the highest number of matches that also satisfy the relation $D(\mathbf{x}_i, \mathbf{x}_K) > 2\varepsilon,\ 1 < i < K - 1$: this condition is needed in order to correctly assign a match to a single *Motif*. Therefore the authors proceed to illustrate a brute-force algorithm to find the *1-Motif*.

3.1.2 Vector Quantization

The theory of Vector Quantization (VQ) can be found in the milestone by Gersho and Gray [27]. The theory comes from a generalization of scalar quantization to vectors, i.e. ordered sets of scalars. For what concern VQ, vectors can be formed by any sequence of samples of

the signal, which can be, among others, a speech signal, a temperature dataset, an image or even a sequence of images, i.e. a video. First of all, we give some basic definitions upon which we will build the entire algorithm; we define a Vector Quantizer, Q, to be a mapping from the k-dimensional Euclidean space $\mathbb{R}^k$ to a finite set $\mathcal{C}$ containing N vectors (*codevectors* or *codewords*) of length k:

$$Q : \mathbb{R}^k \to \mathcal{C}. \tag{3.4}$$

The finite set $\mathcal{C}$, also called *codebook* or *dictionary*, contains vectors who also live in the Euclidean space $\mathbb{R}^k$ and can be formally defined as

$$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_N\}, \quad \mathbf{c}_i \in \mathbb{R}^k \quad \forall i \in \{1, 2, ..., N\}. \tag{3.5}$$

The codebook should be designed to be representative of the entire signal, i.e. the space where vectors live in (formally a subspace of $\mathbb{R}^k$), but it is used through the partition it induces on this space.

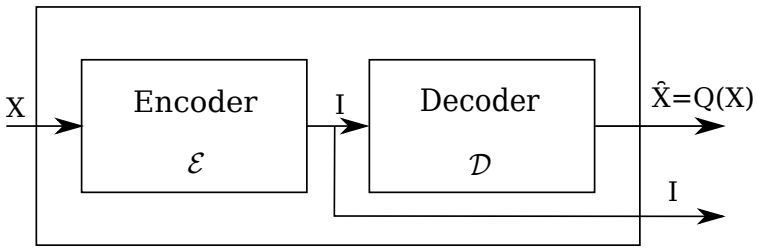


FIGURE 3.1: Encoder-Decoder structure of a Vector Quantizer

It could be more helpful to see the Vector Quantizer as a complete procedure including *encoding* and *decoding*. As shown in Fig. 3.1, the encoder, as already explained, associates to every input vector a codebook index, i; the decoder, on the other hand, uses this codebook index to retrieve the corresponding codeword which represents the input vector, $\hat{\mathbf{x}} = Q(\mathbf{x})$. The quantization error can thus be quantified through the distance between the input vector at time t, $\mathbf{x}_t$, and the assigned codeword $\mathbf{c}_{i\star}$: i.e. $D(\mathbf{x}_t, \mathbf{c}_{i\star}), \quad t \in \mathbb{R}_+$, where $c_{i\star} = Q(\mathbf{x}_t)$.

3.1.2.1 The time-invariant codebook

Having defined how a vector quantizer operates on input signals or, more correctly, on input vectors, it is time to describe how an efficient codebook can be built. The idea is to obtain a good representation of the input vectors using the smallest number possible of entries in the codebook. Such an optimization is obtained though a *partition* of the space where vectors live in, i.e., a set of N regions such that

$$\bigcup_i \mathcal{R}_i = \mathbb{R}^k, \quad \mathcal{R}_i \cap \mathcal{R}_j = \emptyset. \tag{3.6}$$

For each region a vector, or codeword, is chosen as representative for all the input vectors falling inside this region: this entails that, on the decoder side, this vector will be used instead of the input one. It is important to note, with reference to the encoder-decoder structure, that the partition of the space, the codebook, should be shared between the encoder and the decoder, i.e. it must be the same.

We recall, for future use, the definition of *convex set*: a convex set is a set in which, the points of any segment connecting any pair of points belong to the set too; thus a codebook formed by convex regions is said to be *regular* otherwise it is said to be *nonregular*. A subset of the regular quantizers, of particular interest, is constituted by the *polytopal* quantizers, where the regions, not only are convex subsets, but also *polytopes*. To quickly recall the polytope concept we highlight that polytopes are subsets of a space delimited by $(k-1)$-dimensional segments of hyper-planes.

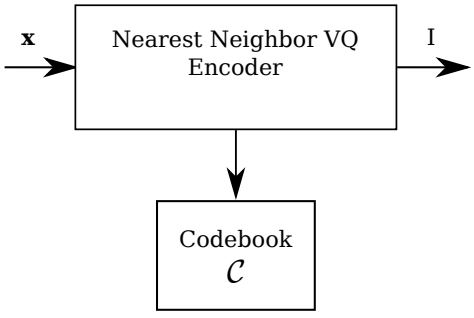


FIGURE 3.2: Structure of a Nearest Neighbor Vector Quantizer Encoder

The codebook construction will change if it is going to preserve a maximum error between input vectors and corresponding codewords or, else, if the requirement is for a maximum allowed number of codewords. The most common type of codebook design is the so-called *nearest neighbor VQ* or *Voronoi VQ*; in fact, very often, these types of VQs are simply referred to as *the* VQs. A Nearest Neighbor quantizer is formally defined as a VQ in which the cells are given by

$$\mathcal{R}_i = \{\ \mathbf{x} : d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j)\ \ \forall j \in \mathcal{I}\ \}, \tag{3.7}$$

where $\mathcal{I}$ is the set of indices labeling the cells. A simple adjustment permits to obtain the necessary mathematical correctness: when a vector lies exactly on a boundary between two cells, the one with the smallest index is chosen. In the Nearest Neighbor VQ any computationally feasible distance measure $d(\cdot)$ can be used: anyway, the most used are the $k-$space usual norms such as L_2, L_1 or the L_∞-norm.

The structure of such a VQ relies completely on a codebook: in defining this codebook we can get rid of the regions and only store the codewords, given that we only need to compute a distance between them and the input vector: its structure can then be depicted as in Figure 3.2.

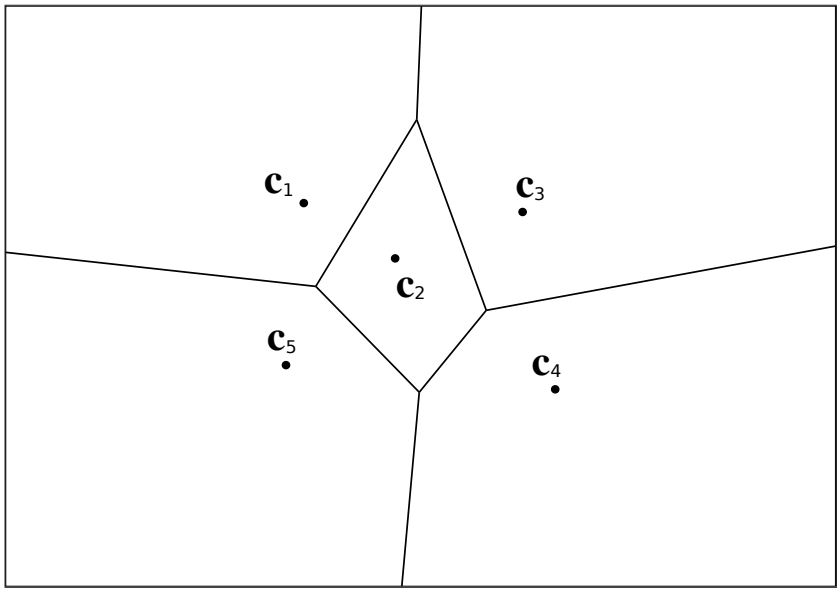


FIGURE 3.3: Partition given by a Nearest Neighbor VQ

If the Euclidean, or L_2, norm is used we can see that that the regions defined by a fixed set of codevectors are polytopes; in fact, we can rewrite the regions as an intersection of hyperplanes

as

$$\mathcal{R}_i = \bigcap_{j,j\neq i} H_{ij}, \tag{3.8}$$

and the hyperplanes H_{ij} are, in this case:

$$H_{ij} = \{ \mathbf{x} : \|(\mathbf{x} - \mathbf{y}_i)\|^2 \leq \|(\mathbf{x} - \mathbf{y}_j)\|^2 \}. \tag{3.9}$$

An insight of how the parition is built is given by Figure 3.3, where the 2-dimensional case with 5 codevectors is repesented, and it is clear that a complete Voronoi diagram is generated.

3.2 RAZOR: mixing Motifs and VQ

The work in [28][29], already exploits the two previous explained techniques, namely, Motifs extraction and Vector Quantization, to build an algorithm able to extract most frequent patterns, classify them and use this classification for compression in a Wireless Sensor Network (WSN): the approach has proven to be both lightweight in computational complexity and accurate in reconstruction.

The algorithm takes advantage of two important tools of VQ, namely *Shape-Gain* VQ and *Mean-Removed* VQ. These useful operations permit to obatin a normalized version of the input vectors, thus retaining only the important features. The two needed parameters, generally called *Offset(O)* and *Gain(G)*, are obtained from the input vector $\mathbf{x} = [x_1, x_2, ..., x_k,]$, through

$$O = \frac{\sum_{i=1}^{k} x_i}{k}, \qquad G = \sqrt{\frac{\sum_{i=1}^{k} x_i^2}{k}} \tag{3.10}$$

and thus the normalized version, $\bar{\mathbf{x}}$, of the vector $\mathbf{x}$ is given by

$$\bar{\mathbf{x}} = \frac{\mathbf{x} - O}{G}. \tag{3.11}$$

The vectors resulting from this normalizaton have thus zero mean and unit gain.

The algorithm specifies also the codebook build-up procedure, to be run on an initial portion, called *training set*, of the signal to be compressed: these initial samples are used to capture the behavior of the signal and to construct a codebook optimized for that signal.

The resulting algorithm, called RAZOR, has two main parts: a first one, in which the Motifs that will make up the codebook are selected, and a second one, in which, after the codebook has been shared with all the nodes, compression takes place.

Motif extraction First, a distance metric, $d(\cdot)$, is selected as in standard VQ, secondly a distance treshold d_{th} is set and finally a maximum allowed size of the codebook, K_{target}, is also set. From the training set sequence, of M_S samples, segments of fixed size k are selected as follows:

$$S^{(i)} = [S_i, S_{i+1}, ..., S_{i+k-1}], \quad \forall i = 1, 2, ..., M_S - k + 1, \tag{3.12}$$

and the normalized versions, $\bar{S}^{(i)}$, are obtained as described earlier. A matrix, DM, is then computed, of which the generic element DM_{ij} is the distance, according to the selected distance measure, between the i-th and the j-th normalized segment:

$$DM_{ij} = d(\bar{S}^{(i)}, \bar{S}^{(j)}). \tag{3.13}$$

Algorithm 1 Codebook build-up procedure

procedure RAZOR MOTIFS EXTRACTION
  for all m,n **do**
    $MM_{mn} \leftarrow I(DM_{mn} < d_{th})$
  $i \leftarrow 1$
  while $DM \neq \emptyset$ and $k \leq K_{target}$ **do**
    for all m **do**
      $MC_m \leftarrow \sum_n MM_{mn}$
    $i \leftarrow \arg\max_m(MC_m)$
    $X(k) \leftarrow \bar{S}^{(i)}$
    Delete the l-th rows and columns in DM and MM for which $MM_{il} = 1$
    $k \leftarrow k + 1$
  $K \leftarrow k$

Algorithm 1 is then applied, in order to obtain the codebook, $X(k), \quad k = 1, 2, ..., K$, but also its size, that could be smaller than the limit K_{target}. Due to its computational cost, this procedure is to be done on a high end machine, such as a *sink* operating in a WSN.[1]

Encoding process After the codebook is determined, it is then shared among all the nodes before the proper encoding process, which is illustrated in Algorithm 2. The procedure simply takes as input vectors constituted by k signal samples and find which, among all the codebook entries, has the minimum distance from the current one, according to the selected distance metric.

Algorithm 2 VQ Compression

```
1: procedure RAZOR COMPRESSION
2:     d_min ← ∞
3:     for all X(k) ∈ CB do
4:         X_R(k) ← X(k)G + O
5:         if d(S^(i), X_R(k)) < d_min then
6:             X̂^(i) = X_R(k)
7:             d_min = d(S^(i), X_R(k))
8:     Return k
```

Application of the proposed algorithm, as it is, has failed revealing what were the peculiarities and thus the needs of biometric signals: the length of the pseudo-period does not let the procedure to build a dictionary (which has fixed size) able to be representative for the entire signal. In fact, the first part of the algorithm could, and should, be skipped given that we already know the recurrent patterns: at least where they begin and where they end.

3.3 Codebook based compression algorithm for biometric signals

Observing typical biometric signals, such as those in Figure 2.2, in wich we can see ECG and a PPG traces, it is clear that there is indeed a *recurrent pattern* and that a method to locate

[1] In Algorithm 1 the symbol $I(\cdot)$ is used to represent the indicator function, that is, $I(E) = 1$ if E occurs, 0 otherwise

these occurences is needed . In fact, the biometric signals considered in this thesis present high frequency components, or peaks, which delimit portions of the signal with similar behavior.

If the location in time of such peaks are found by some method, the signal could be segmented in portions which share a very similar behavior: the idea is that we already know the most recurrent pattern and we could skip the learning phase. For what concern the ECG, some of the works consider, as in the medical field, the P-QRS-T sequence of waveforms to be the recurrent pattern to track: this seemed, since the beginnning, a bad choice given the difficulty to determine the extremes of such a sequence. In this work, it was chosen, instead, to consider as recurrent pattern the signal samples that are between a peak and the following one.

In the following we briefly discuss methods for finding peaks in an ECG trace, for which a very ample literature exists.

3.3.1 The R-peak location problem

That of finding the peak location for an ECG trace has been, since long ago, an important issue: first of all because the exact location let other algorithm to calculate the Heart Rate (HR), which is one of the most important medical parameters.

Works such as [8] presents a detailed review a large number of, continuely proposed, methods: the article focuses on *software* QRS detection as nowadays, multi-purpose processors capable of very fast computations, have become the standard. The algorithms are compared based on accuracy of detection and on the computational complexity required: they are thus classified in high, medium and low complexity classes. Among those in the low class, one prominent approach was chosen as detection method for our compression algorithm: the *Mathematical Morphology*-based method, proposed by Trahanias [30]; already used in [31] for the purpose of QRS detection on low-end devices, it seemed initially a very attractive method for its properties of being lightweight and parameter-free. Later, the method was abandoned in favor of more powerful and targeted algorithms.

3.3.2 Photoplethysmography and other signals

Much less literature exists for other types of signals that are becoming available nowadays on cheap and low end end devices, such as *photoplethysmography* (PPG) obtained through the use of photodiodes on the wrist, as already mentioned. These signals are also linked to the cycle of blood or other vital cycles and thus present the same recurrence as discussed above for the ECG; they need, however, different approaches for what concern peak identification. The PPG, for example, only needs a highpass filtering and then a tresholding, if no major disturbances take place.

Chapter 4

The algorithm

This chapter will be devoted to the detailed explanation of the proposed algorithm, with the description of each functional block and the choices made during its design. A list of parameters will also be presented along, with values assigned to them and behavior of the performances varying these parameters.

4.1 Structure

As anticipated in Chapter 3, the proposed algorithm uses fundamental concepts from Motifs extraction and Vector Quantization through the use of normalized waveforms. The main difference is that the concern of not losing important features of the signal, led to the idea that a static codebook would have entailed too high a loss of clinical information. A possible improvement to be developed was that the codebook could have been maintained online by adding (and maybe removing) codewords on demand.

First of all, the basic structure is represented in Figure 4.1: its main blocks are described in the following sections, with detailed explanation of their functions. It is worth noting, however, that the basic structure is formed by two main parts: a preprocessing stage, devoted to detecting points of segmentation and the proper encoder part, which assigns indices and maintains the codebook.

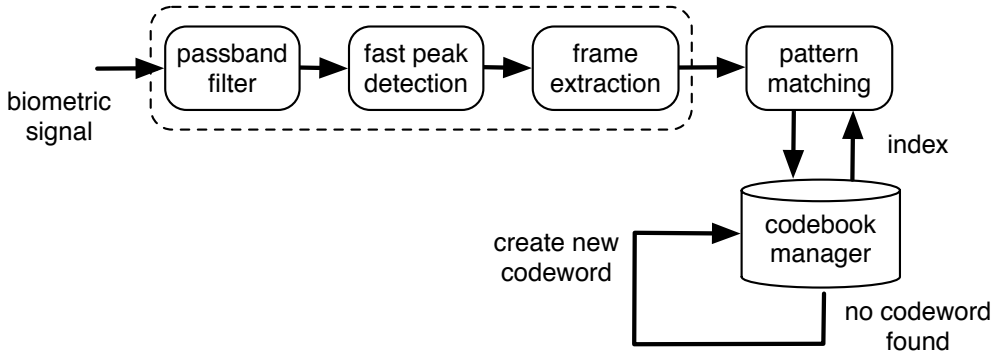


FIGURE 4.1: Basic structure of the proposed algorithm

4.1.1 Bandpass filtering

First a bandpass filter is applied to the signal as it enters the encoder, in order to remove undesired noise; this stage can also be performed by some hardware or firmware embedded in the device. In fact, with some knowledge about the signal we are about to encode, it is possible to avoid various types of noise.

For an ECG signal, for example, a possible choice could be the one suggested by [32], where a third order Butterworth filter is implemented in the bandpass $f \in [8, 20]$ Hz. For the PPG signal a different band should be used, in particular, the PPG does not present the high frequency component that are typical of the QRS coplex and thus a highpass filter, which removes the baseline wander, should be sufficient.

Although this stage seems not instrumental to the encoding, it is helpful in shaping the waveform and thus in reducing the number of useless accesses to the codebook: the proposed algorithm has, indeed, a consevative approach and try to not lose atypical waveforms which could carry important diagnostic information.

4.1.2 Peak detection

As briefly anticipated in Chapter 3, a fundamental step is identifying the points of segmentation of the signal: in most common biometric signals these could safely be taken as the most high

frequency components, or peaks. The ECG waveform has these points, clearly, in the R-peaks, the PPG signal could be segmented in the points of minimum (or maximum) volume of the capillaries, which correspond to the peaks of higher (or lower) intensity detected by the photodiode.

The largest part of the literature is devoted to the identification of the QRS complex (and the R-peak) in an ECG signal, for reasons that are connected with the importance of this signal and the organ it is generated from. Recently some works are also exploring the peak identification in PPG signal, pushed by the advent and diffusion of small and cheap wristbands able to collect it, and the possibility of estimating, from this signal, the Heart Rate with high accuracy [4].

ECG signal For what concerns the ECG signal, being the target wearable and so, battery-operated devices, low computational complexity is a necessary property: the first choice was *Mathematical Morphology* (MM), whose application to QRS detection was first suggested by Trahanias [30]. This technique has two major advantages: it is lightweight in terms of operations needed and it does not need parameters dependent on the ECG features. Mathematical Morphology originated in the image processing field where it was originally used to identify shapes in images; in this use, no prior knowledge on the shapes is needed.

Basically, the procedure for time sequences, works on small windows of the signal in which it selects maximums and minimums among the samples: further elaborations extract the so-called *peaks* and *valleys* of the signal. The lightweight and universal nature convinced the authors of [31] that it was suitable for wearable applications to be used in Body Area Networks (BAN).

Analyzing the MM procedure, we have omitted that a parameter that could be set does in fact exists: the size and the shape of the window that will span the signal. This window, $e(k)$, called *structuring element*, affects the two basic operations of MM, namely *erosion* and

dilation, defined in [31] as follows:

$$\text{Dilate:} \quad f \oplus e(n) = \max_k (f(n-k) + e(k));$$

$$\text{Erode:} \quad f \ominus e(n) = \min_k (f(n+k) - e(k));$$

where k spans the structuring element; however, it is proven that the choice of $e(k)$ does not influence so much the performances of the MM detector proposed in the paper.

Later, this approach was left behind mainly because other methods, more targeted, proved to be more efficient ed effective. In particular, the work in [33] proposes a fast and reliable QRS detector, with the aim of implementing it on portable and wearable devices. Development and application of this algorithm demonstrated it verifies the claims.

It takes advantage of knowledge about the ECG signal, in particular of its frequency band: basically, it resembles the Pan-Tompkins algorithm structure since it includes bandpass filtering, squaring, two overlapping moving averages and the tresholding. The authors of [33] provide also optimized values, for the numerous parameters of the algorithm, when applied to the ECG signal.

Other types of signal A slight modification of the previously explained fast QRS detector, let to exploit it also for photoplethysmography signal; but for other types of biometric signals (blood pressure, respiratory signal), research has to be done, in order to provide the community with energy efficient and accurate algorithms for the detection of peaks.

4.1.3 Frame extraction

This block simply takes as input the positions of the peaks, as provided by the previous block and outputs the signal samples between subsequent peaks, to form a *frame*. Frames, as anticipated, are considered to be the building blocks of the signal, being, as already shown, the recurrent patterns we are searching for. Frames obtained by this procedure are thus passed to the subsequent block.

4.1.4 Codebook manager

The *codebook manager* constitutes the most important block of the proposed algorithm as it provides the proper encoding process. It is to remember that the final goal is compression, but procedures based on Vector Quantization, let to obtain also *classification*: this is a major advantage of this algorithm, whose possible other applications are, surely, many and diverse.

First of all, a method is needed to quantify the *distance* between frames which are, unfortunately, of different lengths: in fact, length should not affect dissimilarity between frames as dilation and compression of the shapes are normally derivating from the different rythms of human body activities.

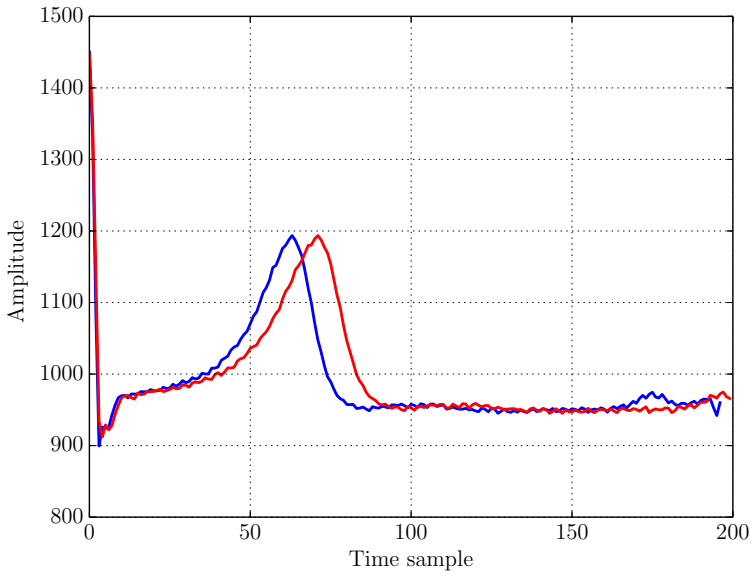


FIGURE 4.2: DTW main drawback: according to DTW the two plotted sequences are very similar

Dynamic Time Warping Although the first thought was for *sampling rate change* techniques, we initially explored the new promising and exciting field of *Dynamic Time Warping* (DTW). The theory of DTW has been studied and deeply analyzed by Keogh and his group [34][35] and improved versions of this approach have also been developed [36].

The DTW was originally developed for speech signals: the same sound of a language can appear, in a time sequence, in a dilated or compressed form and the goal was to set up a method able to overcome this issue. The time warping, exploiting pattern matching techniques coming from the *data mining* field, can efficiently locate and compare sequences, even though stretched in time.

Application of this techniques failed for a simple reason: the excessively time adaptation permitted shapes to move along the time axis, e.g., to the time warping technique, frames with different positioned T-wave in ECG signal, were considered very similar. Figure 4.2 better clarifies this behavior: from a clinical perspective the shift of a T- ot P-wave can have particular implications on the diagnosis.

Multirate processing At this point it was clear that, what was needed, was a method to *uniformly* stretch the frames, given that, variations due to normal change in the ryhtm of body activity, imply uniform stretching in time. Sampling rate conversion, made by *multirate signal processing*, was then reconsidered. Literature on this topic is ample and comprehensive and reviews such as [37] permit to have a comprehensive and global understanding.

The more efficient and straight way to undestand how resampling works, is to see it as a sequence of *reconstruction* and *sampling* processes. This means that the samples are, first, interpolated to a continous domain and then a new sampling occurs. From this point of view, it is clear that much of the accuracy depends on the way interpolation is operated: for the proposed algorithm, a classical *linear interpolation* was considered, for its lightweight character. A wide variety of interpolation functions, however, exists in the literature, among which we cite: linear, quadratic, cubic and splines.

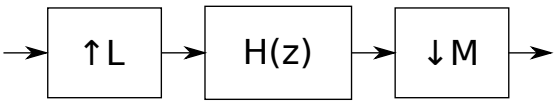


FIGURE 4.3: Building blocks for a general resampling process

The universal structure is depicted in Figure 4.3: a resampling process by a fractional factor, M/L, can always be viewed (and built) by a sequence of an upsampler, followed by the lowpass

filter $H(z)$ and then by a downsampler. According to the proposed algorithm, L is the number of samples we want to resample the frame to and M is the number of samples that make up the original frame.

From an implementation point of view, the main problem with this approach is that the sampling rate alteration factor is continuosly varying, given that the length of the frame arriving at the encoder block, is not known. The main drawback of such a mode of operation, is thus the error introduced by the resampling procedure: in fact, resampling comes from interpolation techniques in which we are just *inventing* values, that the original sequence did not possess. This implies that, to control the resampling error, an analysis of the signal (both in time and frequency) should be done. In this work, the effect of the resampling error is only quantified empirically.

After the resampling has taken place, we are left with a frame of N samples. The codebook manager applies, then, a procedure very similar to the RAZOR algorithm but with a little modification: first the resampled frames are normalized through the process explained in Equation 3.11 and the values of O and G are stored. Subsequently it calculates the distances according to the chosen distance function between the current normalized frame and all the codewords of the codebook, denormalized using O and G. The one with the smallest distance is then selected.

Now, if this minimum distance is less than a given treshold ε, then the corresponding codeword is chosen as representative for that frame and its index is outputted; otherwise, if the minimum distance is greater than ε, the current frame, in its normalized form, is inserted in the codebook.

This procedure can be better understood looking at Algorithm 3, where it is clearer how the codebook is built at runtime during the encoding process.

It should be noted that, theoretically, any type of distance measure can be chosen although two of them were identified as those which provide the best results, namely L_2- and L_∞-norm.

Algorithm 3 CB Compression

procedure CODEBOOK-BASED ONLINE COMPRESSION
  $d_{min} \leftarrow \infty$
  $O = \text{offset}(\mathbf{x})$
  $G = \text{gain}(\mathbf{x})$
  $\mathbf{x}_N = (\mathbf{x} - O)/G$
  for all $\mathbf{c}(i) \in$ Codebook **do**
    $\mathbf{x}_R \leftarrow \mathbf{c}(i)G + O$
    if $d(\mathbf{x}_N, \mathbf{x}_R) < d_{min}$ **then**
      $d_{min} \leftarrow d(\mathbf{x}_N, \mathbf{x}_R)$
      $i^\star \leftarrow i$
  if $d_{min} \leq \varepsilon$ **then**
    return $i^\star$
  else
    add $\mathbf{x}_N$ to Codebook

4.2 Transceiver structure

The algorithm explained above transcended the trasmitter-receiver paradigm, for a clearer explanation: we introduce it now, assigning previous explained blocks to each of the two roles.

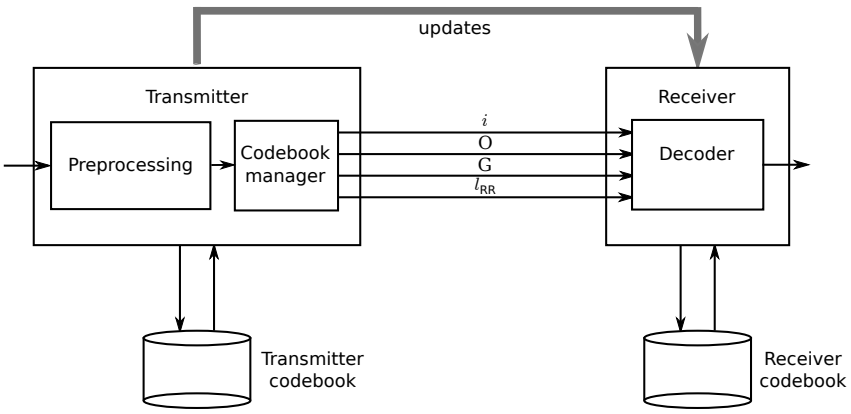


FIGURE 4.4: Transceiver structure of the proposed algorithm

A new structure of the algorithm, which takes into consideration the transmitter and receiver roles, is illustrated in Figure 4.4. As it can be seen, all the previously explained blocks belongs to the transmitter which has the role of receiving the frames as input and managing the codebook. What is sent by the encoder, through the transmitter, to the decoder, is a set of four values: the index of the codebook at which the representative codeword can be found

(i), the values of Gain and Offset as calculated from Equation 3.10 (O and G) and finally the original length of the frame (l_{RR}).

The decoder, for its part, takes as input these four values and uses them to retrieve the codeword and to denormalize it, both in amplitude and time, to obtain the reconstructed signal. It should be noted that we usually have much more computational power avilable at the decoder thus, more powerful resampling functions could be employed such as splines.

What is new, and will be of concern of future developments, is the grey arrow labeled *updates* in Figure 4.4: in fact, it should be clear that, to obtain a correct decoding process, both the codebooks of transmitter and receiver have to be kept synchronized. To achieve this, when the encoder discovers a frame that has to be added to the codebook as new codeword, it has to communicate this *update* also to the decoder by sending the entire codeword as it is.

This update process is the main weak point of the proposed algorithm, but it is heavily dependent on the protocol used for transmission and so it could be alleviated by proper design of the communication interfaces. The design of a suited and reliable communication protocol should be the next step towards the development of the proposed compression approach, with the aim of implementing it on real devices.

Chapter 5

Results

5.1 Databases

Two databases were used for testing the algorithm, both found at the Physionet website [38]: namely Physionet MIMIC II and MIT-BIH Arrythmia. The MIT-BIH databse has a long history as a testbed for all the numerous QRS detection algorithms proposed up to now: it conatains 48 traces of ECG signal (each for a different patient), sampled at 360 samples/s and 11 bits of resolution. The MIMIC II, on the other hand, contains also other types of signals, such as PPG, Blood Pressure (invasive methods), and Respiration Signal, which is a derived signal; these signals were passed through a resampling process which outputs, at the end, 125 samples per second.

In particular, two types of biometric signals were chosen, as they are the most typical nowadays: the ECG, available, until today only in the hospital wards, and now also with the use of some commercial chest bands, and the PPG signal, collected by a large number of commercially available wristbands.

5.2 Size of the codewords

Because of the choice of being conservative in order to not lose important clinical information, the codebook size was left unbounded in this initial development of the algorithm: its behavior constituted, in fact, topic of investigation.

As a first step, we considered the effect that different values of the length of the codewords have on the interpolation error: in fact, the frames extracted from the signal have to be resampled to a fixed number of samples, from now on, N. To determine empirically the optimal value of N, a large number of frames, from different ECG and PPG traces, were resampled to N samples and then, resampled back to their original length: the maximum distance has then been calculated, between the samples of the original and the doubled resampled trace.

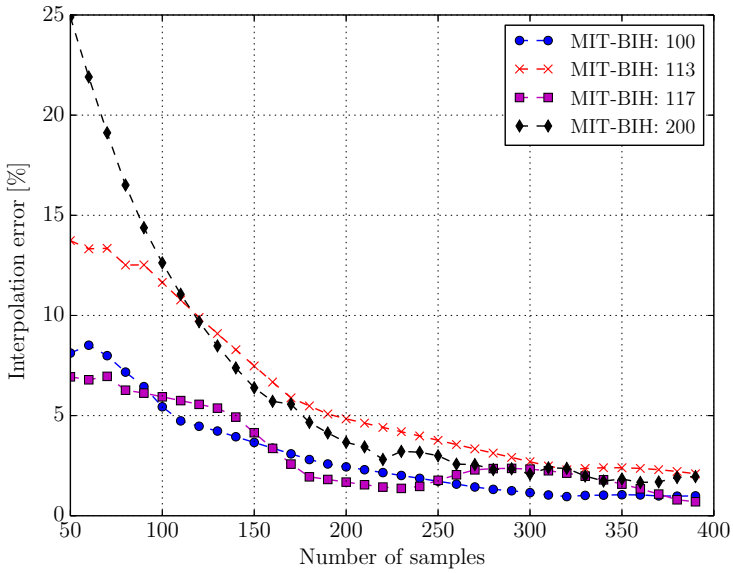


FIGURE 5.1: Behavior of interpolation error

Figure 5.1 shows the maximum error for four ECG traces of the MIT-BIH Arrythmia database: the maximum error is expressed as a percentage of the mean peak-to-peak amplitude (R and Q wave was considered) where the maximum was chosen among the considered signals. For the MIT-BIH, a tradeoff that let to limit computational complexity and the interpolation error to remain under 5% is thus $N = 200$ samples; value that can be used with the MIMIC II database too, and that will be used in the following sections.

5.3 Competing compression algorithms

To evaluate the algorithm, its performance has been compared against other compression algorithms: in particular, a *Discrete Wavelet Transform* (DWT) based algorithm, a *Discrete Cosine Tranform* (DCT) based technique and a *Piecewise Linear Approximation* (PLA) based compression method have been chosen.

For what concerns DWT, the algorithm described in [13] was selected, given its reliability and lightweight approach. The procedure (labeled DWT-LW) selects the Wavelet coefficients based on energy tresholds and uses an efficient method to further encode the indices of the selected coefficients. A second version (labeled DWT-TOL) was obtained through the use of Inverse DWT, and an iterative check on the maximum difference between original and reconstructed samples.

The DCT algorithm was developed in two versions: a first approach (labeled DCT-TOL) calculates the DCT coefficients and uses an Inverse DCT to check how many coefficients are needed to meet a requirement on the maximum distance between the original and reconstructed sequences; a second approach (labeled DCT-LW) selects coefficients based on energy tresholds, as the DWT does, that are to be set as parameters.

As a representative for the PLA class of compression algorithms, it was selecteded the one which has proven to be the best in terms of computational complexity and compression capabilities, in the environmental datasets, namely *Lightweigth Temporal Compression* (LTC).

No algorithms such as AZTEC or CORTES were chosen, because of their high computational complexities, but, most importantly because of their being too much targeted: in fact, for many of them, it is impossible an application on signals different from the ECG.

The performance metrics to be evaluated are three: *energy consumption*, *reconstruction fidelity* and *compression capability*. Since each compression algorithm has its own parameters that are to be set, to compare the performace of each the following strategy is adopted: the compression algorithms are first run on the same biometric trace, spanning a large range of the parameters, and the values of the considered metrics are then compared in a feasible interval.

5.4 Hardware architecture

To reliably compare the algorithms through the energy comsumption metric, a hardware architecture must be selected as a reference testbed: the energy spent for operating the compression is, indeed, highly dependent on the way basic operations are physically implemented. To be foresighting, but in a realistic way, an ARM based architecture was chosen and, in partcular, the Cortex M4 processor, which implements a *Floating Point Unit* (FPU).

addition	multiplication	division	comparison
1	1	14	1

Table 5.1: Cortex M4 operations cycles

Thus, the various algorithms can take advantage of the high efficiency intorduced by the FPU module to do real numbers calculations, as can be seen in Table 5.1, which values come from [39].

Processors such as the Cortex M4 have been considered, in the past, to be too much expensive in terms of energy consumption: in fact, the tradeoff between computational power and energy saving had always been paying much more attention to the second. Nowadays architectures such as ARM, continously under development, introduced a very high computational power by remaining low on the energy consumption side. From [39] [40] is also possible to retrieve the energy spent per CPU cycle, which is 32.82 pJ.

Considering also the transmission of the signal, an energy consumption for transmission is also calculated: as a reference testbed for this energy, a very common Bluetooth transceiver is considered, namely the Texas Instruments CC2541 Bluetooth LE wireless interface. Data used to calculate the transmission energy can be found in [41].

5.5 Performance evaluation

To be clear, it is helpful to give some definitions that will be used throughout what follows. The *Root Mean Square Error* (RMSE) between vector $\mathbf{x}$ and vector $\mathbf{y}$ (of dimension N) is

defined to be:

$$\mathrm{RMSE}(\mathbf{x},\mathbf{y}) = \sqrt{\frac{\sum_{n=1}^{N}(x_n - y_n)^2}{N}} \tag{5.1}$$

The RMSE is used, instead of *Percentage RMS Difference* (PRD), often used to assess performaces of ECG compression algorithms, for two reasons: first, because we also use other signals and second, because the assumption on its definition is quite always wrongly made as it is dependent on the mean of the signal.

With the term *compression efficiency*, it is indicated the ratio between the bits needed to send the signal, as it is sampled by the medical device, and the bits needed to transmit the compressed signal, namely

$$\mathrm{CE} = \frac{\text{Number of bits of the original signal}}{\text{Number of bits of the compressed signal}} \tag{5.2}$$

We remark that, in the calculation of CE for our proposed algorithm, we made a very limiting assumption: we considered the values to be transmitted (O, G, i and l_{RR}) as represented with the same number of bits used for the original samples.

The energy consumption is calculated through the use of four fundamental operations which are always implemented in the hardware architecture of every processor: addition, multiplication, division and comparison. Evaluating the number of cycles needed for each operation and knowing the energy spent per cycle, we obtained the total amount of energy spent to carry out the compression.

5.5.1 ECG

Next, we show quantitative results for the considered signal compression algorithms, detailing their energy consumption, compression e reconstruction fidelity performance. As a first set of results, in Figure 5.2 we plot the RMSE as a function of the compression efficiency for the selected algorithms. Here, the RMSE has been expressed as a percentage of the average peak-to-peak amplitude (R-to-Q-peak amplitude) of the input signal. From Figure 5.2 one can observe that there is no single best algorithm: some schemes perform best in RMSE al

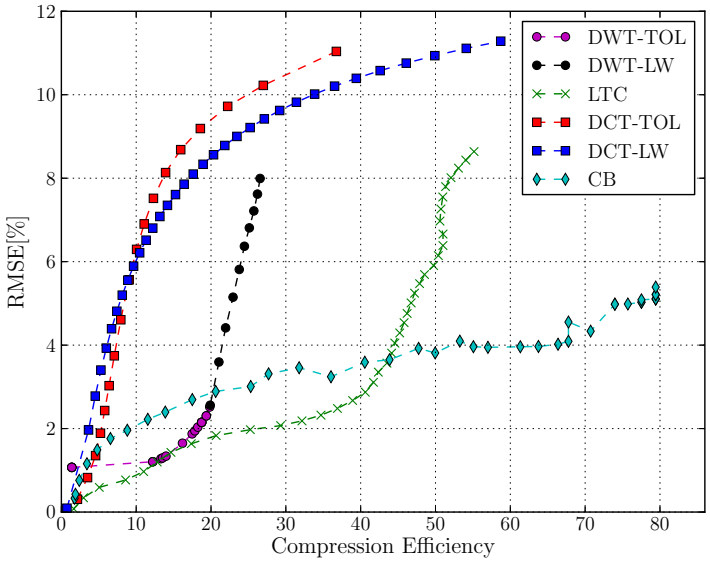


FIGURE 5.2: ECG Signal: RMSE *vs* Compression Efficiency

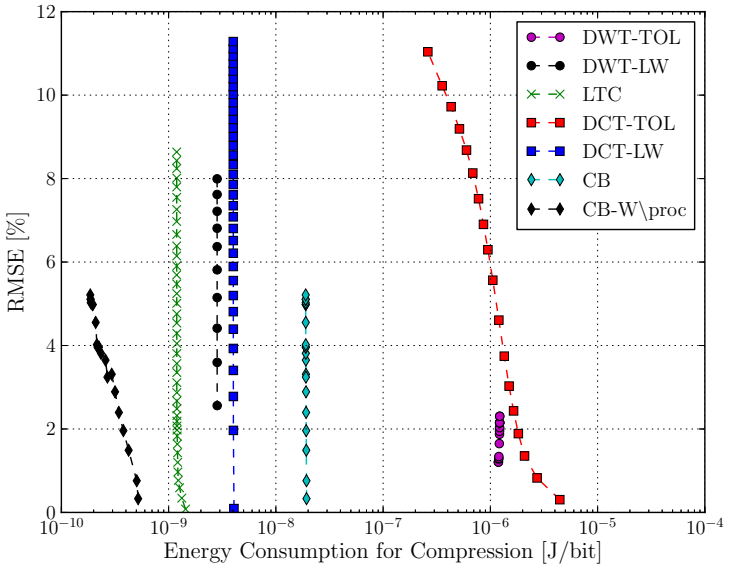


FIGURE 5.3: ECG Signal: RMSE *vs* Energy Consumption for Compression

low values of compression efficiency while others at higher values and they differ, notably, in terms of maximum achievable compression.

At low compression efficiencies, DWT and LTC provide the best reconstruction fidelity. DCT performs a little worse because its coefficients are not able to capture, at a high degree, the behavior of the signal. The proposed method (labeled CB) performs satisfactorily at low compression ratios and, most notably, maintains the RMSE quite small (see its trend for an increasing compression efficiency) while reaching the highest compression ratios, i.e., as high as 80 times for the considered signal and a 30% greater that the maximum compression efficiency reacheable by LTC.

In Figure 5.3, it is shown the RMSE against the energy required by the considered 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 numerous inverse transforms at the encoder side. LTC is energy efficient and is in fact a good candidate, despite its simplicity (piecewise linear approximation lies at its core). The codebook based proposed algorithm performs in the middle, between DWT/DCT (in their heavy variants) and LTC. However, we underline that the energy consumption of the codebook based scheme is dominated by the preprocessing stage, as depicted in Figure 4.1. To see its effect, the energy consumption of the algorithm, without considering this stage, is also plotted as *CB-W\proc*.

As anticipated, the codebook based scheme, without preprocessing, is the most energy efficient algorithm. Note that filtering is always performed on digital devices, to remove measurement artifacts, and peak detection is also very often calculated to extract relevant signal features such as the instantaneous heart rate. Given this, the energy consumption associated with the required preprocessing functions may not be a problem, especially if these functions are to be implemented anyway.

In Figure 5.4 it is shown the compression efficiency against the *total* energy consumption, that is, the sum of the energy spent for compression and for transmission of the compressed signal. From this plot it is clear that the total energy consumption is dominated by the transmission

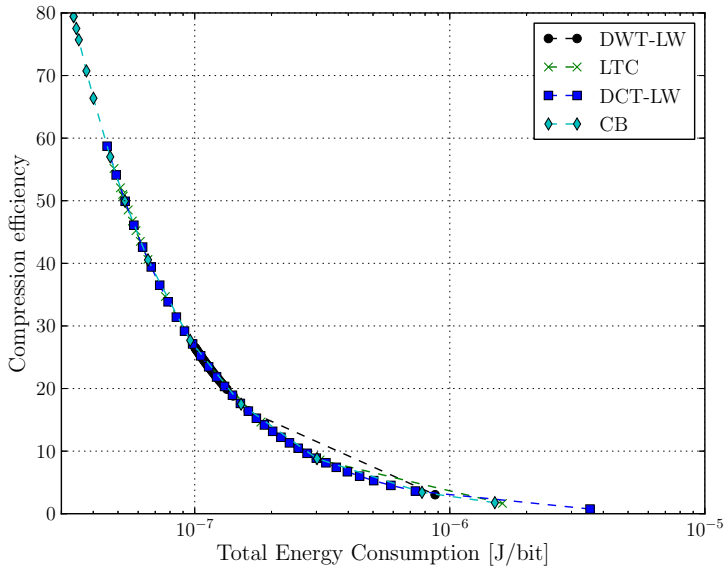


FIGURE 5.4: ECG Signal: Compression Efficiency *vs* Total Energy Consumption

task: as a consequence, the compression efficiency plays a key role in reducing the energy drained, but different alorithms do not show noticeable differences.

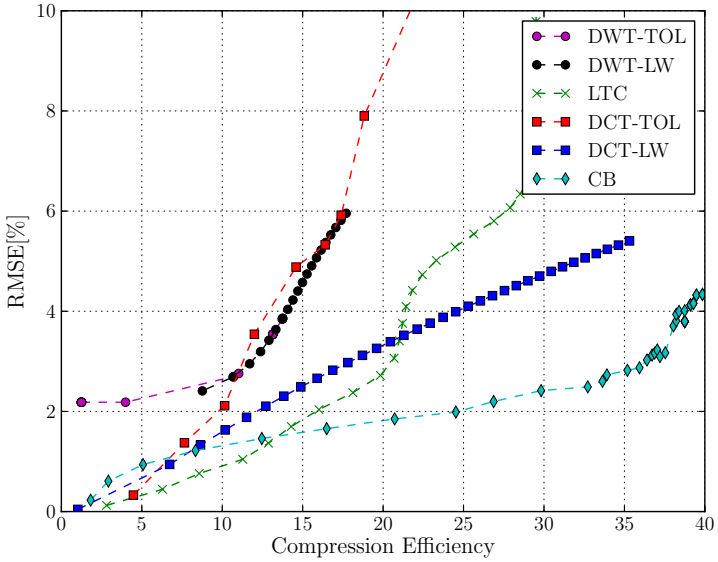


FIGURE 5.5: PPG Signal: RMSE *vs* Compression Efficiency

5.5.2 PPG

Performance evaluation for the photoplethysmogrphy signal is also carried out: a first attempt of applying the algorithm as it was applied to the MIT-BIH ECG signals, highlighted another parameter of the algorithm; in fact, signals taken from the Physionet MIMIC II database (from which the PPG traces were taken) have a rate of 125 samples per second, which is pretty low for medical signals: this means that there will be much less samples, also per period, with respect to the ECG signals of the MIT-BIH. All this led to a maximum value of compression saturating at approximately 20 times: a simple, but effective work-around, permitted to double the maximum compression reachable to 40 times.

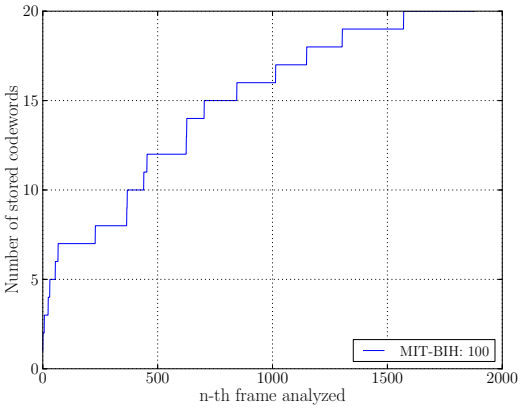
Basically, it can be observed that the PPG signal is even more repetitive than the ECG: its behavior changes very slowly even when clinical episodes appear, thus the codewords, to be stored in the codebook, were not chosen to be samples between adjacent peaks, but a peak was skipped every two.

This strategy should be regarded as an adaptation that is always made when applying the algorithm: in fact the approach is strongly dependent on the distribution of the redundancy in time and thus, before it can be implemented on hardware, a study of the signals to be compressed should be carried out.

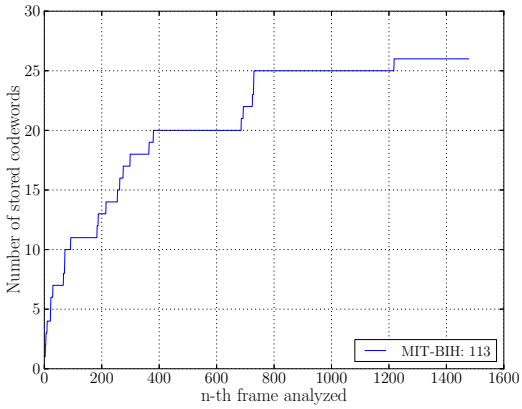
From Figure 5.5 it can be seen that the algorithm, after the adaptation has been made, outperforms the competing techniques, particularly in the region of high compression ratios. It is possible to see that the points accumulate around high values of compression efficiencies, this means that the algorithm is particularly robust when taken to extreme performance: it does not show any tendency to grow incontrollably, in the RMSE value, for high compression ratios.

5.5.3 Codebook growth

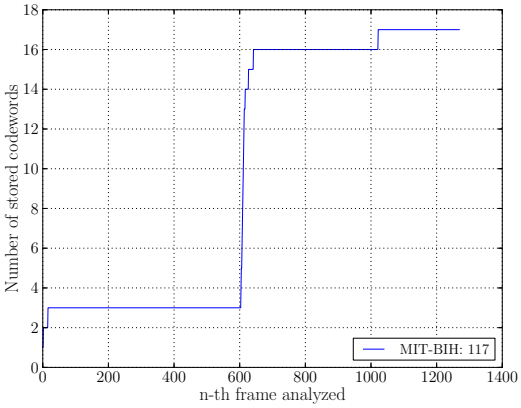
It is interesting, also, to understand the dynamics of the codebook itself: it is expectable that its size would increase in the initial phase of the waveform acquisition, during which



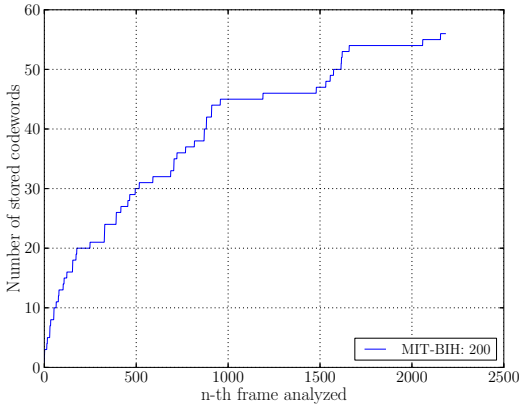
(A) MIT-BIH 100 trace



(B) MIT-BIH 113 trace



(C) MIT-BIH 117 trace



(D) MIT-BIH 200 trace

FIGURE 5.6: Codebook growth in time

it should be learning the waveform typical behavior, and then it should be converging to a dimension value, with some new codeword added now and then, i.e., when some strange activity takes place. It should also be noticed that, for the traces used here it is not possible to have a comprehensive view on the codebook size behavior, beacuse their length, 25 minutes of recording, is not sufficient: a longer testbed with real devices is going to provide, surely, a better insight.

Looking at Figures 5.6 it can seen that this is indeed the codebook general behavior, but only for some traces. The database used is the MIT-BIH and the signals are all ECG traces, the signals were encoded using a tolerance of 10% of the average peak-to-peak amplitude of the signal (meaning that noise peaks were not considered). In Figure 5.6c, as an example, there is a clear arrythmia episode taking place around the 600th encoded frame, and also Figure 5.6b

has two episodes in which the signal behavior changes abruptly, namely around the 360th and 700th ancoded frame.

From this set of results, it is clear that some type of initial learning, i.e, a *learning phase*, could be helpful also to the proposed algorithm. In fact, allowing the codewords to be updated according to the statistics of the signal, should lead to a more rapidly converging behavior of the codebook, thus improving the compression efficiency. This will be, however, of concern to future development.

Chapter 6

Conclusion

Although some work remains to be done, the fundamental aspects of the proposed algorithm were explained and analazyed; the various parameters and their effects were listed along with the reachable performance. The main aspects that were not possible to deeply study include the theoretic fundations of the interpolation error and its effects on the codebook building process. In fact, the codewords were separated imposing a distance treshold ε, resulting in a certain representation error. On top of the latter, it should be added the error due to resampling, which affects the encoding part of the scheme. A future work will include this analysis and provide, possibly, some other cutting-edge method for comparing different length sequences.

In this work a novel approach to the design of a codebook-based compression algorithm, which rely on Motif extraction and Vector Quantization, has been proposed. This approach was found superior to existing schemes in terms of compression efficiencies, whilst retaining very good reconstruction accuracies at the decompressor. We remark that, in addition to its inherent compression capabilities, the proposed scheme enable further processing functions such as classification, learning and statistical computations. In fact, the codewords (patterns or Motifs) in the dictionary and the related number of uses per codeword can be utilized to asses statistical properties of the corresponding signals, including their most relevant features.

Future developments will be centered on these facts and on the study of self-adapting dictionaries based on Neural Networks (NN), most notably Self-Organizing Maps (SOM). These, in addition to learning codewords at runtime, as it is already done, will also be capable of adapting them according to changes in the signal statistics. Finally, a complete work can also be done which can include signals coming from other subject-related sources such as wearable 9-axis motion tracking sensors.

Appendix A

Multirate DSP: resampling

In the following a quick recall of the Multirate DSP resampling technique is exposed. The concepts illustrated in what follows were, for the most part, taken from [37]. First of all we define the building blocks of any Multirate digital system, namely the *upsampler* and the *downsampler*.

(A) Upsampler (B) Downsampler

FIGURE A.1: Symbols for upsampler and downsampler

The downsampler of *rate* M is simply described as a block which retains a sample every M of the input sequence and loses all the other samples. The upsampler of *rate* L, on the other hand, is a block which inserts $L-1$ zero-valued samples between a sample and the next of the input sequence. Their pictorial representation can be seen in Figure A.1.

(A) Interpolator (B) Decimator

FIGURE A.2: Symbols for interpolator and decimator

Two functionals are defined, which respectively decreases and increases the number of samples of an input sequence, by making use of a transfer function $H(z)$. The *decimator* is constitued

by a lowpass filter followed by a downsampler and the *interpolator* is made up by an upsampler followed by a lowpass filter as shown in Figure A.2. The filter $H(z)$ seems to have different purposes in the two configurations: in the decimator it is used to avoid aliasing which can be introduced by the downsampler, in the interpolator it eliminates images of the spectrum introduced by the upsampler, thus shaping the signal spectrum. Later it will be seen that in fact, these two functions, can be unified in a single lowpass filter with transfer function $H(z)$.

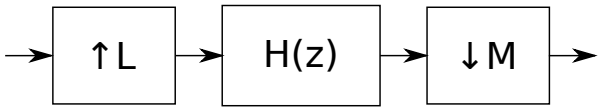


FIGURE A.3: Fractional resampling process obtained thorugh decimator and interpolator

The blocks previously decribed operate on integer rates, namely M and L. If a rational changing factor is needed, it is possible to put an interpolator and a decimator in sequence: first the sampling rate is increased by a factor L and then it is decreased by a factor M thus obtaining the rational factor L/M. It is worth noting that the order of the operations cannot be changed, otherwise the interpolator loses information about the original sequence and it is not able to operate correctly. From such a setting, it can be seen that the lowpass filters operate at the same rate, besides doing the same function: the final fractional rate changer can thus be build as in Figure A.3.

A different point of view can be beneficial for a more theoretical approach. The fractional resampling process can be viewed as a *sampling after reconstruction* process. Considering the basics of interpolation as known, one can consider the application of interpolation to a given input sequence, thus obtaining *reconstruction*, and then the application of a classical sampling block, which operates with sampling period $T = M/L$. The result is a sequence whose sampling rate is L/M times the original sampling rate.

Exploiting the way the low pass filter $H(z)$ is implemented, it is possible to obtain various types of resampling schemes, namely linear, quadratic and splines to cite some of the most famous: their performance can heavily affect their computational complexity.

Bibliography

[1] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From Today's INTRAnet of Things to a Future INTERnet Of Things: A Wireless- and Mobility- Related View," *Wireless Communications, IEEE*, vol. 17, pp. 44–51, December 2010.

[2] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric Sensing," *Philosophical Transactions of Royal Society*, vol. 370, pp. 176–197, November 2012.

[3] R. Yousefi, M. Nourani, S. Ostadabbas, and I. Panahi, "A Motion-Tolerant Adaptive Algorithm for Wearable Photoplethysmographic Biosensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 670–681, March 2014.

[4] Z. Zhang, Z. Pi, and B. Liu, "TROIKA: A General Framework for Heart Rate Monitoring Using Wrist-Type Photoplethysmographic Signals During Intensive Physical Exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, pp. 522–531, February 2015.

[5] "Zephyr BioHarness3." `http://zephyranywhere.com/products/bioharness-3/`.

[6] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking," *ACM Transactions on Sensor Networks*, vol. 11, pp. 1–34, November 2014.

[7] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 32, pp. 230–236, March 1985.

[8] B.-U. Kohler, C. Hennig, and R. Orglmeister, "The Principles of Software QRS Detection," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, pp. 42–57, January 2002.

[9] M. Elgendi, B. Eskofier, S. Dokos, and D. Abbott, "Revisiting QRS Detection Methodologies for Portable, Wearable, Battery-Operated, and Wireless ECG Systems," *PloS ONE*, vol. 9, p. e84018, January 2014.

[10] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *IOP Physiological Measurement*, vol. 28, pp. 1–39, February 2007.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] X. C. Qu and Y. Zhang, "A Compression Algorithm for ECG Data Using Variable-Length Classified Template Sets," in *Conference: 2014 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 856–859, IEEE, 2014.

[21] Y. Noh and D. Jeong, "Implementation of ECG Template Matching Compression Algorithm for Mobile Healthcare Device," *Journal of Information Processing and Management*, vol. 4, no. 3, pp. 260–268, 2013.

[22] C.-C. Sun and S.-C. Tai, "Beat-based ECG compression using gain-shape vector quantization," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1882–1888, 2005.

[23] A. Ramakrishnan and S. Saha, "ECG compression by multirate processing of beats," *Computers and biomedical research*, vol. 29, no. 5, pp. 407–417, 1996.

[24] S.-G. Miaou and J.-H. Larn, "Adaptive vector quantisation for electrocardiogram signal compression using overlapped and linearly shifted codevectors," *Medical and Biological Engineering and Computing*, vol. 38, no. 5, pp. 547–552, 2000.

[25] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in *Proceeding of the 2nd Workshop on Temporal Data Mining*, SIGKDD '02, (New York, NY, USA), pp. 53–68, ACM, 2002.

[26] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," in *Proceedings of the 8th ACM SIGMOD*

Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03, (New York, NY, USA), pp. 2–11, ACM, 2003.

[27] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, Massachusetts: Kluwer Academic Publishers, 1992.

[28] M. Danieletto, *Design and Evaluation of Compression, Classification and Localization Schemes for Various IoT Applications*. PhD thesis, University of Padova, January 2014.

[29] 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, 2013.

[30] P. E. Trahanias, "An Approach to QRS Complex Detection Using Mathematical Morphology," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 2, pp. 201–205, 1993.

[31] F. Zhang and Y. Lian, "QRS Detection Based on Multiscale Mathematical Morphology for Wearable ECG Devices in Body Area Networks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, pp. 220–228, August 2009.

[32] 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, pp. 478–484, February 2008.

[33] M. Elgendi, "Fast QRS Detection with an Optimized Knowledge-Based Method: Evaluation on 11 Standard ECG Databases," *PLoS ONE*, vol. 8, pp. 1–18, September 2013.

[34] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, pp. 358–386, March 2005.

[35] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *Third Workshop on Mining Temporal and Sequential Data*, pp. 22–25, 2004.

[36] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, pp. 561–580, October 2007.

[37] P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 56–93, 1990.

[38] "Physionet website." `http://physionet.org/`.

[39] "ARM Cortex-M4 Processor Specifications page." `http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php`.

[40] ARM, "ARM Cortex-M4 Technical Reference Manual," 2010.

[41] S. Kamath and J. Lindh, *Measuring Bluetooth® Low Energy Power Consumption*. Texas Instruments, 2012.