Information Theory Concepts Applied to the Analysis of Rhythm in Recorded Music with Recurrent Rhythmic Patterns

Martín Rocamora[1*], *AES Associate Member*, **Pablo Cancela**[1], **Luiz W. P. Biscainho**[2], *AES Member*
(rocamora@fing.edu.uy) (pcancela@fing.edu.uy) (wagner@smt.ufrj.br)

[1] *Universidad de la República, Uruguay*
[2] *Universidade Federal do Rio de Janeiro, Brazil*

Repeating patterns are essential in music understanding and data compression. This paper applies information theory concepts to rhythmic analysis of recorded percussion music. Downbeat detection is addressed via lossy coding of an accentuation feature under rate-distortion criteria, assuming the correct alignment produces the simplest explanation for the data. The resulting description is suitable to related tasks, e.g. assessing performances' complexity and estimating the number of different rhythmic patterns played.

0 INTRODUCTION

Often a parallel is drawn between data compression and computational learning [1]. The argument runs as follows: the more we are able to compress the data, the more we have learned about its underlying regularities. Similarly, it is also common to draw a connection between data compression and complexity assessment [2, 3]. Simply put, data compression captures the amount of structured information present in a certain phenomenon, therefore the compression ratio can serve as a measure of the complexity of the data. This idea has been applied in a myriad of disciplines [4, 5, 6], including music modeling [7, 8, 9].

Part of music understanding can be seen as a problem of finding repeated patterns, and thence structure [10, 11]. Ultimately, data compression can be tailored to the problem of explicitly finding structure through repeated patterns in the data under analysis [12]. Some recent works address the analysis of symbolic representations of musical pieces through general purpose text compression techniques [13] and point-set compression algorithms [14].

Consequently, information theory—and particularly, data compression—stands as an appealing framework for music modeling. Source coding is a mapping from a sequence of symbols from an information source to a sequence of alphabet symbols, such that the source symbols can be exactly recovered (*lossless coding*), or recovered with some distortion (*lossy coding*). In his foundational work on information theory in 1948, Claude Shannon established the limits to possible data compression [15]. His source coding theorem states that it is not possible to losslessly compress the data using an average number of bits per symbol (i.e. coding rate) smaller than the *entropy* of the source. In this context, *entropy* is the expected value of the information contained in each message, where information is defined as the cologarithm of the symbol probabilities.

Information theory aims at providing a measure of the amount of information conveyed by the data, which can be interpreted as the length of its most compact description. In this approach, the messages to be encoded are supposed to be outcomes of a known random source, whose characteristics determine the encoding. Therefore, given a random source of known characteristics, we are interested in the minimum expected number of bits per symbol to transmit a message from the source through an error-free channel.

For lossy source coding, Shannon introduced and developed the theory of source coding with a fidelity criterion, also called *rate-distortion* theory, which provides the theoretical foundations for lossy data compression [15, 16]. In practice, when we have a continuous source we are not necessarily interested in exact recovery, but only in approximate recovery within a given tolerance. Hence, a distortion measure is introduced to account for the average information loss. The problem of coding is then formulated as determining the minimal number of bits per symbol, as measured by the coding rate, so that the source can be approximately recovered without exceeding a given

[*] To whom correspondence should be addressed Tel: +598-2711-0974; Fax: +598-2711-7435; e-mail: rocamora@fing.edu.uy

distortion value. Rate-distortion theory has been studied by the information-theory community for more than fifty years [17, 18]. Its concepts are an essential component of many lossy compression techniques and standards, and have been successfully applied to lossy coding of speech, high-quality audio, images, and video [18, 19]. Nevertheless, the application of a lossy source coding scheme to the description of music and the analysis of its structure remains, to the best of our knowledge, virtually unexplored.

The ubiquitous application of digital technology to music distribution/storage has fueled a new multidisciplinary field of research during the last few decades: Music Information Retrieval (MIR) [20]. It focuses on the processing of music-related digital data (such as editorial metadata, scores, lyrics, and audio), and the development of methodologies to process and understand such data. Most of the proposed methods rely on audio content, and different kinds of information are extracted by means of signal processing techniques [21]. The extraction of musically meaningful rhythm-related information from audio recordings is a core task in MIR [22, 23], with applications in digital audio workstations for music editing and processing, DJ-mixing software and hardware products, and intelligent organization and navigation over large music collections.

The attribute of rhythm is of central importance in music. It concerns the way in which the musical events are arranged in time, grouped and organized into structures and patterns. There is a broad agreement on the importance of rhythmic patterns as structural elements in music [24]. From Western Africa traditions to European folk dances, repetitive rhythmic patterns are the core of rhythmic/metrical structures. In modern music theory, metrical structure is described as a regular pattern of points in time, hierarchically organized in metrical levels of strong and weak beats [25]. *Beats* specifically refer to the pulsation of the perceptually most salient metrical level, and are further grouped into *measures* or *bars*. The first beat of each measure is called the *downbeat*. The metrical structure itself is not present in the audio signal, but is rather inferred by the listener through a complex cognitive process [24].

The main motivation of this work arises from a novel idea: to recast the downbeat detection task as a data compression problem. Different possible alignments of the beats within the rhythm cycle are evaluated, and a parsimony criterion is used to select the one corresponding to the downbeat. The hypothesis is that the correct alignment will allow for a simpler explanation of the data than the misaligned ones. To this end, one adopts a lossy compression framework based on rate-distortion theory, suitable to the continuous data source analyzed: an accentuation feature function directly computed from audio. In this way, a sort of music structure analysis problem—in its minimal expression—is formulated in terms of rate-distortion theory. It turned out that the description obtained is well suited for addressing other related tasks, namely complexity assessment of performances and estimation of the number of different rhythmic patterns found in a given recording.

The rest of the document is organized as follows. The next section reviews the rate-distortion theory and describes the methods applied. In Section 2 the proposed approach to deal with music audio recordings is presented. Some experiments are conducted using an existing dataset of percussion music recordings, which provides a suitable scenario for bringing into focus the rhythmic aspects of music, without the need of considering the interplay with other music dimensions, such as melody and harmony. The experiments and respective results are reported in Section 3. The paper ends with a discussion on the present work, including promising directions for future research.

1 Rate-distortion theory

An introduction to *rate-distortion* theory usually recalls that the description of a real number requires an infinite number of bits, thus a finite representation of a continuous random variable X can never be perfect [26]. However, having defined some sort of evaluation measure, one can try to quantify how good the representation is. This is accomplished through the introduction of a measure of *distortion*, d, to describe the distance between the random variable and its representation. By allowing a certain degree of distortion, the amount of bits used in the representation can be reduced. In communication theory, this becomes the problem of determining the smallest number of bits per symbol (as measured by the average bit rate R) that must be transmitted through an ideal channel so that the system input signal is reconstructed at the receiver with an average distortion not higher than D. Thus, a communication system involving an encoder and a decoder can be formulated based on rate-distortion.

1.1 Encoding

Consider such a rate-distortion encoder/decoder system applied to a random variable X. Let $X^n = X_1, X_2, \dots, X_n$ be a sequence i.i.d $\sim p_X(\mathrm{x})$, $\mathrm{x} \in \mathscr{X}$. This source sequence $X^n \in \mathscr{X}^n$ is represented by the encoder as an index $f_n(X^n) \in \{1,2,3,\dots,2^{nR}\}$. The decoder represents X^n by an estimate $\hat{X}^n \in \hat{\mathscr{X}}^n$. A $(2^{nR}, n)$-rate distortion code can be defined, which consists of an encoding function,

$$f_n : \mathscr{X}^n \rightarrow \{1,2,3,\dots,2^{nR}\}, \tag{1}$$

and a decoding or reproduction function,

$$g_n : \{1,2,3,\dots,2^{nR}\} \rightarrow \hat{\mathscr{X}}^n. \tag{2}$$

The decoded sequence $g_n(f_n(X^n)) = \hat{X}^n$ is a quantized version of the original source sequence X^n according to a scheme that is optimal for a given distortion measure.

1.2 Vector quantization

If we are given R bits per symbol to represent source X, the problem is to find the optimum reproduction points which minimize a distortion measure, i.e. design a vector quantizer Q that maps an Euclidean space of dimension k, $\mathscr{R}^k$, to a finite *codebook* $\mathscr{C}_M$ with M *codevectors* $\hat{\chi}_m \in \mathscr{R}^k$.

$$Q : \mathscr{R}^k \rightarrow \mathscr{C}_M = \{\hat{\chi}_1, \hat{\chi}_2, \dots, \hat{\chi}_M\}. \tag{3}$$

Codevector $\hat{\chi}_m$ has an associated reconstruction region or cell

$$\mathscr{R}_m = \{\mathrm{x} \in \mathscr{R}^k \mid Q(\mathrm{x}) = \hat{\chi}_m\}. \tag{4}$$

The encoder is completely specified by the partition of $\mathscr{R}^k$, and the decoder is completely specified by the codebook. Given a distortion measure $d(\mathrm{x}, Q(\mathrm{x}))$, the mean distortion value achieved by the system is computed as

$$D = \int_{\mathscr{R}^k} d(\mathrm{x}, Q(\mathrm{x}))\ p_X(\mathrm{x})\ \mathrm{dx}, \tag{5}$$

where $p_X(x)$ is the probability density function of X.

Two simple properties are useful to find a proper vector quantizer. First, given a set of reconstruction points $\{\hat{\chi}_i\}$, the distortion is minimized by mapping each element of X^n to the closest of them (the *nearest-neighbor* condition). The set of nearest-neighbor regions with respect to the distortion measure is called a *Voronoi* or *Dirichlet* partition. Then, given a certain partition, the reconstruction point for each region should be selected in order to minimize D, which is accomplished by selecting the centroid of the region as the reconstruction point (the *centroid* condition). The *generalized Lloyd* algorithm for designing a vector quantizer is based on these properties [27, 28].

1.3 Generalized Lloyd algorithm

The generalized Lloyd algorithm [28] is an iterative algorithm that starts with a certain set of reconstruction points and finds the optimal reconstruction regions as the nearest-neighbor regions with respect to the distortion measure. Then, new optimal reconstruction points are chosen as the centroids of the reconstruction regions, and the procedure is repeated. In this way, the expected distortion decreases at each iteration, and the algorithm converges to a local minimum in a finite number of iterations. A stopping criterion has to be applied, for instance not surpassing a minimum amount δ of distortion decrease between iterations. The algorithm is summarized in Algorithm 1.

Algorithm 1 Generalized Lloyd algorithm

```
step 1: $m = 1$, initial codebook $\mathscr{C}_1 = \{\hat{\chi}_1\}$ and distortion $D_1$
step 2: given codebook $\mathscr{C}_m$ find codebook $\mathscr{C}_{m+1}$ by
  2.a finding nearest-neighbor regions $\{\mathscr{R}_m\}$ to partition $\mathscr{R}^k$
  2.b setting reconstruction points $\{\hat{\chi}_{m+1}\}$ as centroids of $\{\mathscr{R}_m\}$
step 3: compute distortion $D_{m+1}$ for new codebook $\mathscr{C}_{m+1}$
if ($D_m - D_{m+1} > \delta$) then
    goto step 2
```

It is worth noting the close relationship between the generalized Lloyd algorithm and the k-means clustering algorithm [29]. The latter also repeatedly finds the centroid of each set in the partition, and then re-partitions the input according to the closest centroids. But the main difference is that k-means clustering operates on a discrete set of points instead of a continuous region. Thus, repartitioning the input means simply determining the nearest centroid to the finite set of points, whilst the generalized Lloyd algorithm actually partitions the whole space into regions. If the input is a finite set of points, both algorithms are equivalent.

1.4 Distortion measure

The distortion function $d(\mathrm{x}, \hat{\mathrm{x}})$ measures the cost of representing symbol x by symbol $\hat{\mathrm{x}}$. It can be regarded as a mapping $d : \mathscr{X} \times \hat{\mathscr{X}} \to \mathscr{R}^+$, from the set of pairs of the source alphabet and the reproduction alphabet into non-negative real numbers. To measure the distortion between sequences x^n and $\hat{\mathrm{x}}^n$, the average of the per–symbol distortion of the elements of the sequence can be computed:

$$d(\mathrm{x}^n, \hat{\mathrm{x}}^n) = \frac{1}{n} \sum_{j=1}^{n} d(\mathrm{x}_j, \hat{\mathrm{x}}_j). \tag{6}$$

A very common distortion function is the *squared-error* distortion. Given $\mathrm{x}, \hat{\mathrm{x}} \in \mathscr{R}^k$, such that $\mathrm{x} = [x_1, x_2, \ldots, x_k]$ and $\hat{\mathrm{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k]$, it can be defined as the squared 2-norm of the difference between symbols normalized by k,

$$d(\mathrm{x}, \hat{\mathrm{x}}) = \frac{1}{k} \|\mathrm{x} - \hat{\mathrm{x}}\|_2^2 = \frac{1}{k} \sum_{i=1}^{k} (x_i - \hat{x}_i)^2. \tag{7}$$

1.5 Operational rate-distortion curve

The relationship between rate and distortion can be described by a *rate-distortion function*, $R(D)$, that determines the set of possible achievable points in the rate-distortion trade-off for a certain statistical source class [19]. In order to derive such bounds the source has to be properly characterized, but this can be troublesome for complex sources, such as audio and video signals.[1] Besides, the bound provided by a theoretical rate-distortion function gives no constructive procedure for attaining that optimal performance.

Instead, a practical quantization scheme can be examined, and the best operating points of this particular system can be searched for. If all possible quantization choices for that system are considered for a certain source (described by a statistical model or a training set), an *operational rate-distortion curve* [19, 30] can depict the distortion achieved by the best encoder-decoder pair designed for each rate. Its points are said to be operational because they are all achievable with the chosen quantization implementation for the available data. This curve allows to identify the best achievable operating points as well as to differentiate them from those that are sub-optimal or unachievable. When we can make the search among a fixed and discrete set of parameters, each combination of parameters gives a certain R-D pair, producing a curve of individual admissible operating points. In this case, the convex hull of the set of operational points defines the boundary between achievable and non-achievable performances [19], as shown in Figure 1.

1.6 Optimization

Within this rate-distortion framework, given a source with a certain distribution and a distortion measure, we seek to establish what is the minimum expected distortion at a particular rate, or equivalently, what is the min-

[1] A closed form can be found for $R(D)$ in special cases, e.g. the Gaussian source with squared-error distortion, or the binary memoryless (Bernoulli-p) source with Hamming distortion [26]. For other distributions, numerical methods must be applied [19].

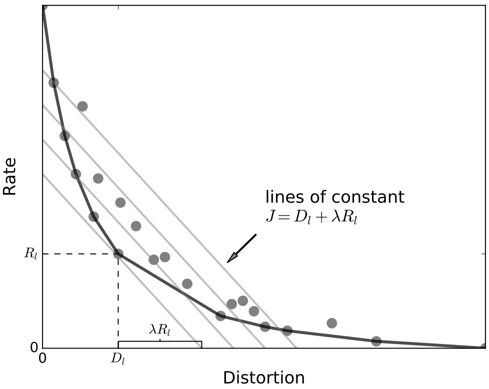


Fig. 1. Schematic diagram of an operational rate-distortion curve, with the operational points and their convex hull. The discrete version of the Lagrangian optimization is also depicted.

imum rate description required to achieve a certain distortion [26]. This can be posed in the form of constrained optimization problems. That means considering either a cost function D with constrained rate $R \leq R_c$, or conversely a cost function R with constrained distortion $D \leq D_c$.

The solution of a constrained optimization problem can often be found by using the Lagrangian method, which minimizes an unconstrained cost function that is the sum of the original objective function and a term that incorporates the constraint and a real multiplier $\lambda \geq 0$, known as Lagrange multiplier. This is a well known technique for problems where the cost function is continuous and differentiable. Yet, when the operational rate-distortion curve is considered, one can apply a discrete version of the Lagrangian [19] that is able to find and optimal solution as long as there exists a point in the convex hull that meets the required constraint. Let l be an index used to denote the operational points on the convex hull of the curve such that as l increases, the rate decreases and the distortion increases. The discrete optimization problem can be formulated as

$$\underset{l}{\text{minimize}} \;\; J = D_l + \lambda R_l. \tag{8}$$

For a particular value of λ, the Lagrangian rate-distortion functional J is minimized as follows. Find the point on the convex hull that intersects, among all line contours with a given J value (i.e. with slope equal to $-1/\lambda$), that one with the smallest J value. Figure 1 illustrates the procedure.

Each choice of λ can lead to the selection of a specific optimal point in the rate-distortion trade-off. In particular, minimizing J when $\lambda = 0$ is equivalent to minimizing the distortion, whereas minimizing J when $\lambda \to \infty$ is equivalent to minimizing the rate. Intermediate values of λ determine intermediate operating points. Finding the λ value that provides an optimal solution at the required rate can be done using approaches such as the bisection search [19].

2 Application to musical rhythm analysis

The proposed approach is based on the idea of describing the rhythmic information of a complete music performance by using a rate-distortion coding scheme. We expect that by studying the coding trade-off between the number of bits per symbol and the amount of distortion we gain some insight into the characteristics of the performance.

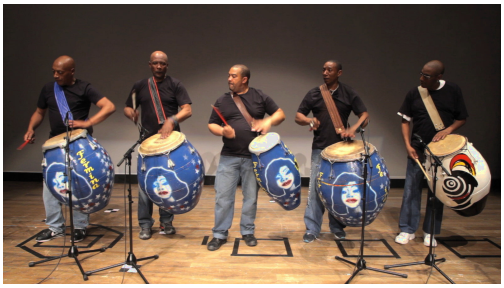


Fig. 2. Candombe players at the recording session. From left to right: *repique*, *piano*, *chico*, *piano*, and *repique*.

2.1 Percussion music analysis

In order to focus on the rhythmic aspects of music without the need of considering other music dimensions, the present study deals with percussion music. In particular, it examines a certain type of Latin American music of African origin: the *candombe* drumming, one of the most defining traits of popular culture in Uruguay [31]. Like in other musics of the Afro–Atlantic tradition, such as Afro–Cuban, the *candombe* shows repetitive rhythmic patterns. Its rhythm cycle, comprising four beats, can be subdivided in sixteen pulses or *tatums*. The rhythm results of the interaction among rhythmic patterns of three drums of different size and pitch, called *chico*, *repique* and *piano*. The drum–head is hit with one hand bare and the other holding a stick, as shown in Figure 2. The stick is also used to hit the shell when playing the *clave*, a pattern used for temporal synchronization. Examples of *clave*, *chico* and *piano* patterns are shown in music notation in Figure 3 (top).

Since its pattern is the most informative on both beat and downbeat locations, the analysis method proposed is tailored towards the *piano* drum, i.e. the largest and lowest sounding of the three drums. Actually, the *piano* drum has two main functions: playing the base rhythm with characteristic one–cycle patterns (*piano base*), and occasional more complex figurations (*piano repicado*), typically one or sometimes two cycles long. The many pattern variants that can be found depend on both the style of each neighborhood and the individual style of the performer.

2.2 Audio feature extraction

An audio recording of a complete music performance is represented using spectral audio features to serve as the primary input source to encode. This is done in two steps.

Step 1: The spectral flux [32] is calculated for the signal under analysis to produce an accentuation feature that emphasizes the onset of notes by seizing the changes in its spectral magnitude along different frequency bands.

Firstly, one calculates the Short-Time Fourier Transform (STFT) of the discrete-time audio signal $s[t]$ as

$$S(u,z) = \frac{1}{T}\sum_{t=0}^{T-1} w[t-uh]\; s[t]\; e^{-j\frac{2\pi}{T}zt}, \tag{9}$$

where u is the signal frame index, z is the frequency bin index, h is a hop size in samples (20 ms) and $w[t]$ is a smoothing Hann window (40 ms). Then, the equal-spaced bins of the STFT are combined into fewer bands whose center frequencies follow the Mel scale so as to better approximate human auditory resolution, i.e. coarser at high frequencies and finer at low frequencies [33]. The magnitude of the Mel-scaled short-time spectra is time-differentiated, and the resulting sequences are half-wave rectified to consider only positive magnitude changes. Summing along the MEL frequency bins z', one obtains the accentuation feature

$$SF(u) = \sum_{z'=0}^{Z-1} H\left(|S(u,z')| - |S(u-1,z')|\right), \tag{10}$$

where $H(x) = \frac{x+|x|}{2}$, denotes half-wave rectification.

In principle, the feature value is high when a stroke has been articulated and close to zero otherwise. But it also carries some information on the type of stroke. For instance, an accented stroke produces a higher feature value compared to a muffled one, since the spectral change is more abrupt and typically encompasses a wider frequency band.

To roughly separate the rhythmic patterns of the different drums, a sub-band filtering is implemented by summing the spectral flux along different frequency bands, as shown in Figure 3. Only the low-frequency band up to approximately 200 Hz—corresponding to the *piano* drum—is used in the reported experiments. A local amplitude normalization is carried out to preserve intensity variations of the rhythmic patterns while discarding long-term fluctuations in dynamics. A p-norm within a local window is applied as

$$\overline{SF}(u) = \frac{SF(u)}{\sqrt[p]{\sum_{v=-\Delta}^{\Delta} |SF(u+v)|^p}}, \tag{11}$$

where p controls the type of norm and Δ determines the window length. A value of $p = 8$ was used in the reported experiments, so that if the feature in the current frame is close to the highest value within the window it is normalized to 1. For the normalization to behave as desired, the Δ parameter must be selected such that several sound events lay within the window. This is implemented by considering Δ to be proportional to the tempo of the performance, i.e. $\Delta = \mathrm{T}\ \tau$, where τ stands for the *tatum* period in samples and $\mathrm{T} > 0$ is an integer value. A value of $\mathrm{T} = 4$ was used in the reported experiments, which corresponds to a window of approximately half a rhythm cycle. Note that the *tatum* period is estimated from manual annotations of the beats, available for the dataset used in the experiments; it could also had been inferred automatically from the audio signal as in [34].

Step 2: The accentuation feature is organized into a feature map. Firstly, the feature signal is time-quantized to the rhythm metric structure by considering a grid of

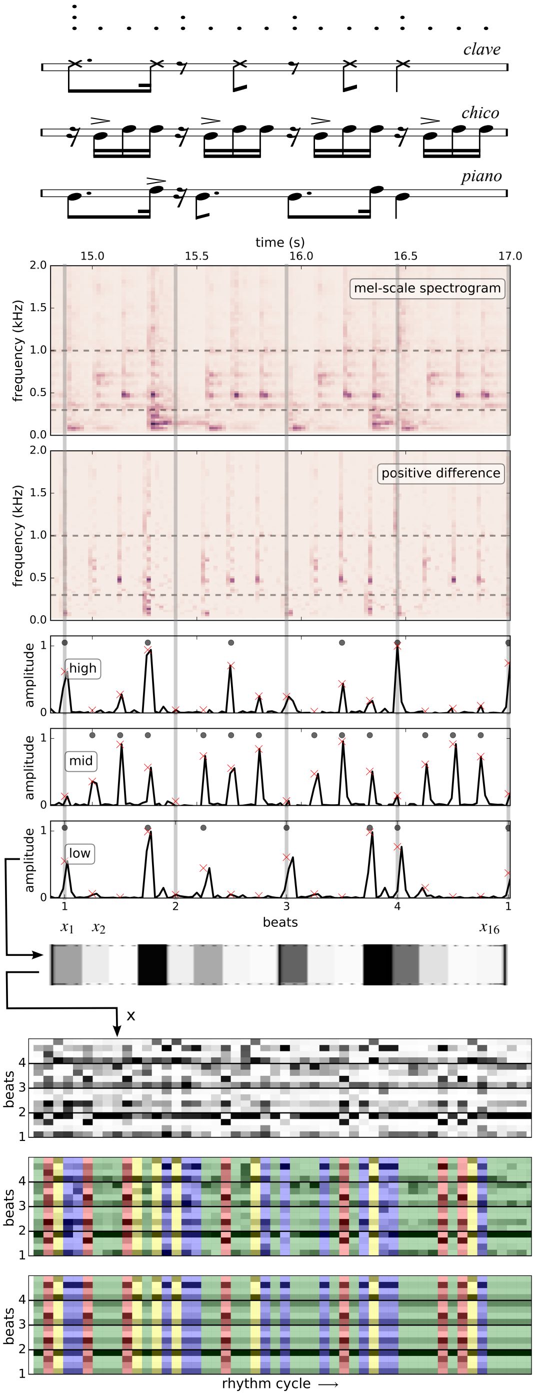


Fig. 3. Example of feature extraction for a synthetic audio signal combining *piano*, *chico* and *clave* patterns, as indicated in music notation. The first two plots are the magnitude of the Mel-scaled short-time spectra, and its half-wave rectified first-order difference. Next, the accentuation feature extracted in three different frequency bands (high, medium, and low) delimited by the dashed lines is shown. Beat locations are indicated by vertical lines. In the accentuation feature plots: time-quantized feature values are denoted by red crosses; the articulated events of each pattern are depicted with dots, and approximately match the peaks of the feature signal. The low-frequency feature signal is used to build a map of rhythmic patterns where feature values are represented as shades of gray (darker colours for higher values), and each cycle-length pattern becomes a column. The three feature maps at the bottom correspond, in descending order, to: the input sequence x^N to be encoded, the clusters shown with colors, and the output sequence $\hat{\mathrm{x}}^N$ of codevectors.

tatum pulses equally distributed within the annotated beats. The corresponding feature value is taken as the maximum within a 100 -ms window centered at the frame closest to each *tatum* instant. This yields 16 -dimension feature vectors whose coordinates correspond to the *tatum* pulses of the rhythm cycle.

Then, a feature map of the cycle-length rhythmic patterns of a performance is obtained by building a matrix whose columns are consecutive feature vectors. An example of such a map, computed for the low-frequency band of a recording in the dataset, is provided in Figure 3 (bottom). The horizontal axis corresponds to the rhythm cycle index, while the vertical axis corresponds to the sixteen *tatum* pulses of a cycle, increasing upwards by convention. This representation enables the inspection of the similarities and differences between patterns, as well as their evolution over time. Note that if a certain *tatum* pulse is articulated for several consecutive rhythm cycles, it will be shown as a horizontal line in the map.

2.3 Proposed rate-distortion method

The feature map of the performance is the primary input source to encode. The resulting input space $\mathscr{R}^k$ has dimension $k = 16$, corresponding to the number of *tatum* pulses in the rhythm cycle. Thus, the input vectors are of the form

$$\mathrm{x} = [x_1, x_2, \ldots, x_{16}]. \quad (12)$$

Since features are normalized, each component x_i takes values in $[0, 1]$. A complete performance of length N rhythm cycles is represented by the sequence

$$\mathrm{x}^N = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N\}. \quad (13)$$

This sequence can also be regarded as a matrix

$$\mathrm{X} = (\mathrm{x}_{i,j}), \quad i \in [1, 16], \quad j \in [1, N], \quad (14)$$

where i is the *tatum* index and j is the rhythm cycle index, which represents the feature map.

The vector quantization is implemented according to the generalized Lloyd algorithm. For this particular case, in which the input is a finite set of points, this corresponds to the k-means clustering algorithm. Therefore, each rhythmic pattern x_j of the performance is clustered to a particular group $\mathscr{R}_m$ and represented by its centroid $\hat{\chi}_m$. Figure 3 shows with different colors the grouping obtained for a codebook of size $M = 4$. The input sequence x^N is represented by the encoded sequence $\hat{\mathrm{x}}^N$ comprising only elements of the codebook $\mathscr{C}_M$.

A distortion value, $d(\mathrm{x}_j, \hat{\chi}_m)$, is computed between every pattern symbol x_j of the sequence and its corresponding codevector $\hat{\chi}_m$, using the squared-error distortion defined in Equation 7. Then, the distortion of the whole input sequence x^N is obtained by averaging the per-symbol distortion, using Equation 6. The bit-rate R of the encoded sequence $\hat{\mathrm{x}}^N$ is computed as

$$R = -\sum_{m=1}^{M} p_m \log_2(p_m) \quad (15)$$

where M is the codebook size and p_m is an estimate of the probability of occurrence of each symbol. The probability estimate p_m is obtained as $p_m = \frac{n_m}{N}$, with $n_m = \#\{\hat{\chi}_m = \hat{\mathrm{x}}_j\}, \forall \hat{\mathrm{x}}_j \in \hat{\mathrm{x}}^N, j \in [1, N]$, where # denotes the cardinality of the set, and thus n_m represents the number of occurrences of the codevector $\hat{\chi}_m$ in the encoded sequence $\hat{\mathrm{x}}^N$, which is normalized by the total length N of the sequence.

The coding process described so far relies on a single parameter, namely the codebook size M. Therefore, an operational rate-distortion curve is obtained by varying the codebook size M and computing the corresponding values for rate and distortion. An example of this type of operational curve is depicted in Figure 4, for the same audio file used in Figure 3. Note that the rate is expressed in bits and the distortion is a mean squared-error value. The number next to an operational point indicates the corresponding value of the codebook size M. The behavior of the rate-distortion curve is as expected: as the codebook size is increased, the distortion diminishes while the rate grows.

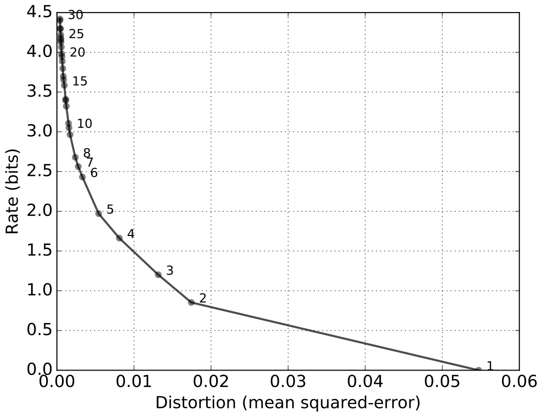


Fig. 4. Operational rate-distortion curve for a real recording.

The k-means clustering is initialized with reconstruction points selected at random, which can have an impact on the obtained clusters. For this reason, in the reported experiments the k-means clustering is repeated 10 times and the best solution is selected according to the overall minimum sum of distances of cluster members to centroids. To further mitigate the initialization effect, the process for computing every point in the curve is repeated 10 times, and the median values for rate and distortion are used.

3 Experiments and results

Three different types of experiments are reported aiming at assessing the usefulness of the proposed approach. Firstly, the operational rate-distortion curves are used to qualitatively characterize drumming performances in terms of their overall complexity. Some possible implications of this method to the description of performance style and player expertise are also discussed. Then, the problem of estimating the number of different rhythmic patterns in a given performance is addressed within the rate-distortion framework. The solution investigated corresponds to selecting an operational point in the curve that adequately balances the rate-distortion trade-off. Finally, in the light of the previous experiments, the last problem addressed aims at identifying which one of the beats corresponds to the downbeat, without using any high-level information about the rhythm except for its four-beat division. By comparing the rate-distortion curves of the different possible alignments of the four beats within the rhythm cycle, it turns out that the correct solution yields the less complex representation for a large part of the available recordings, thus allowing the automatic detection of the downbeat. The underlying rationale for the success of the method as well as its limitations are discussed and illustrated with examples.

3.1 Dataset of audio recordings

The music corpus for this study is a dataset of *candombe* recordings released in a previous work [34].[2] The recordings were produced using professional audio equipment, during various studio sessions [35], in the context of musicological research. The audio files are stereo with a sampling rate of 44.1 kHz and 16-bit precision. The dataset comprises 35 complete performances, totaling over 2 hours of audio. A total of 26 renowned players took part, in groups of three to five drums. The location of beats and downbeats was manually annotated by a music expert.

3.2 Comparison of performance complexity

The operational rate-distortion trade-off will show a different behavior depending on the performance complexity. Firstly, the rate-distortion curve is determined by the number of codevectors needed to properly encode the sequence. For instance, if there are several different rhythmic patterns played, then a small codebook size will not suffice to correctly describe the performance and will necessarily yield a high distortion value. Besides that, there is the issue of how well each group of patterns is represented by a single codevector. The amount of variability of the patterns within a certain group will also contribute to increase the distortion, even for the correct codebook size. For these reasons, the rate-distortion curves of different performances will lie in different regions, simpler performances yielding lower rate-distortion values compared to the more complex ones. This is illustrated in the following experiment.

3.2.1 Experiment 1

Four different complete performances from the dataset were selected and classified by a music expert with regards to the overall complexity of the *piano* drum part. For each recording the low-frequency feature was extracted, the input sequence x^N was constructed using the beat/downbeat labels and the operational rate-distortion curve was computed varying the codebook size from 1 to 30. This is the standard procedure adopted for all the reported experiments. The resulting curves are depicted in Fig. 5, together with an excerpt of the input sequence x^N of each performance. From left to right, the input sequences are sorted in a decreasing order of complexity, according to the judgment of the music expert. Note that the same ordering is evidenced in the operational rate-distortion curves, the more complex performances indicated with darker lines.

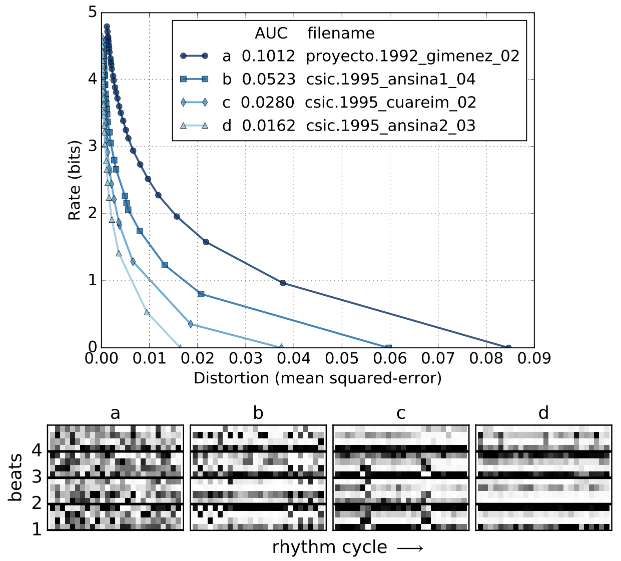


Fig. 5. Comparison of performance complexity. Rate-distortion curve (top) and an excerpt of the input feature sequence x^N (bottom) for four different complete performances from the dataset.

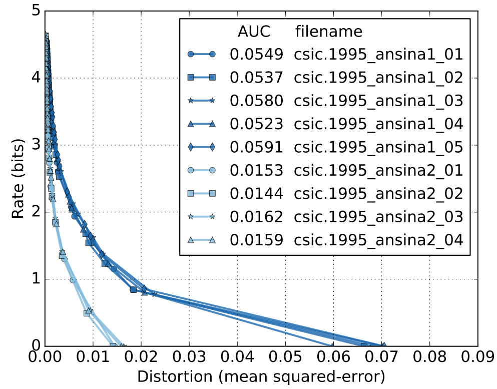


Fig. 6. Comparison of the rate-distortion curves of complete performances corresponding to two different *piano* drum players.

There are many possible ways to characterize the rate-distortion curves and to summarize their behavior into a single number. For instance, the distortion value for the codebook size $M = 1$, i.e. the zero-rate point, preserves the ordering of performance complexity. However, it ignores the behavior of the curve for other codebook sizes. Another option is to compute the area under curve (AUC). To do that, the curve is extrapolated to estimate a cut-off point in the ordinates (with a polynomial fit considering the last 10 values) and the AUC is calculated using the numerical trapezoidal rule for approximating integrals. The AUC values obtained in this way are shown in Fig. 5 and are consistent with the qualitative ordering of the performances.

[2] http://www.eumus.edu.uy/candombe/datasets/ISMIR2015/

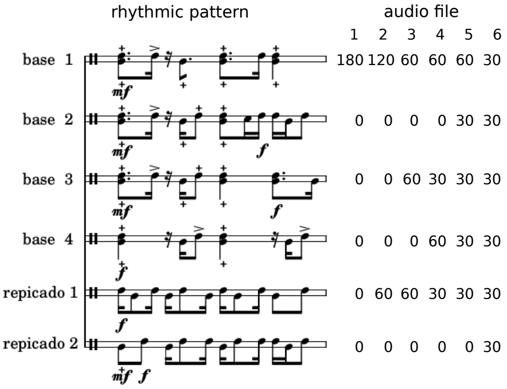


Fig. 7. Rhythmic patterns used in the experiment (left) and number of cycles per rhythmic pattern in each audio file (right).

3.2.2 Experiment 2

It is reasonable to assume that the degree of complexity displayed in a performance is voluntarily controlled by the player, depending on the musical context. At the same time, it can be associated with personal style and expertise.

In order to illustrate these issues, a comparison is carried out considering 9 performances from the same recording session, the *piano* drum played by one performer in five of them and by another one in the remaining four. The rate distortion curves and their AUC values are presented in Figure 6. Two groups of recordings are readily distinguishable, each one corresponding to a different performer. This indicates their personal styles were consistent and clearly different from each other during the whole recording session, which once again matches subjective assessment.

3.3 Estimation of number of rhythmic patterns

The next problem addressed is the automatic estimation of the number of rhythmic patterns in a given recording. This can be useful for detection and classification of rhythmic patterns, performance style comparison and beat/downbeat tracking [36, 34]. Since the feature values are continuous and the rhythmic patterns may exhibit several variations within a recording, the problem can be regarded as finding a good compromise between a concise account of a given performance and a sufficiently precise description of its rhythmic patterns. Within the rate-distortion framework this corresponds to selecting a certain operating point of the trade-off. If a detailed representation is required, then the number of rhythmic patterns (i.e. the codebook size) has to be increased, at the expense of a necessarily longer performance description (i.e. higher rate). This can be posed as an optimization problem which can be solved using the discrete version of the Lagrangian method [19]. But it still requires that one finds the optimal value for λ, a problem tackled in the following experiment.

3.3.1 Experiment 3

If a sufficiently large and representative training set is available, one can search for an optimal value for the Lagrange multiplier λ, in the sense of yielding the correct number of rhythmic patterns for most of the data at hand. This approach is illustrated in the following. A set of rhythmic patterns usually found in *candombe* performances was considered, and audio files that followed them were synthesized. To do that, music scores with the rhythmic patterns were produced in a general purpose music engraving software language, adopting some conventions to represent the different types of strokes. Several sound samples of each type of stroke, recorded by a professional musician, were randomly selected by the synthesis program, which is able to interpret local accents and variations in dynamics.

The music scores of Fig. 7-left represent the six *piano* rhythmic patterns that were used in the experiment, comprising four *base* patterns and two *repicado* patterns. Lower and upper line represent hand and stick strokes respectively and the muffled strokes are indicated with a cross. Six audio files of the same length (180 rhythm cycles) were rendered by gradually incrementing the number of different patterns included, up to a uniform distribution of all of them. Fig. 7-right shows the of number of cycles per rhythmic pattern in each audio file.

Then, the rate-distortion curves were computed and the discrete Lagrangian method was applied to them, i.e. the minimization in equation 8 was performed. For each curve, the λ values that yield the correct number of patterns were looked for, following a grid-search scheme. The grid of values considered is in the range $\lambda = [0.001, 0.05]$ with a step of 0.0001. Fig. 8 shows the rate-distortion curves, along with the extremes of the grid represented with a dashed line. The range of valid λ values for each audio file, i.e. the ones producing the correct number of patterns, is also indicated as a light grayed out region. If the extent of valid λ values among different files is considered, it turns out that the range $\lambda_{[1,6]} = [0.0058, 0.0099]$ yields the correct solution for all files, shown as a darker grayed out region. The next experiment tests this approach with real recordings.

3.3.2 Experiment 4

This experiment tackles the estimation of the number of different rhythmic patterns in real recordings, considering the four complete performances introduced in Fig. 5. For this purpose, the discrete Lagrangian method is applied using a value of $\lambda^* = 0.00785$, the mean of the range $\lambda_{[1,6]}$. This is represented graphically in Fig. 9-top, as lines with slope $-1/\lambda^*$ intersecting each rate-distortion curve. The solutions obtained in this way suggest a number of patterns M of 6, 4, 3, and 2 for the recordings sorted in decreasing order of complexity. The encoding of each performance is presented in Fig. 9-bottom using the corresponding estimate of the number of patterns as the codebook size.

Unlike the previous synthetic experiment there is no ground-truth in this case, but the automatic encodings match the description of the performances provided by the music expert, which, from a musicological point of view, validates the approach.

3.4 Downbeat detection

The last type of experiment recasts the downbeat detection task as a data compression problem. Assuming the lo-

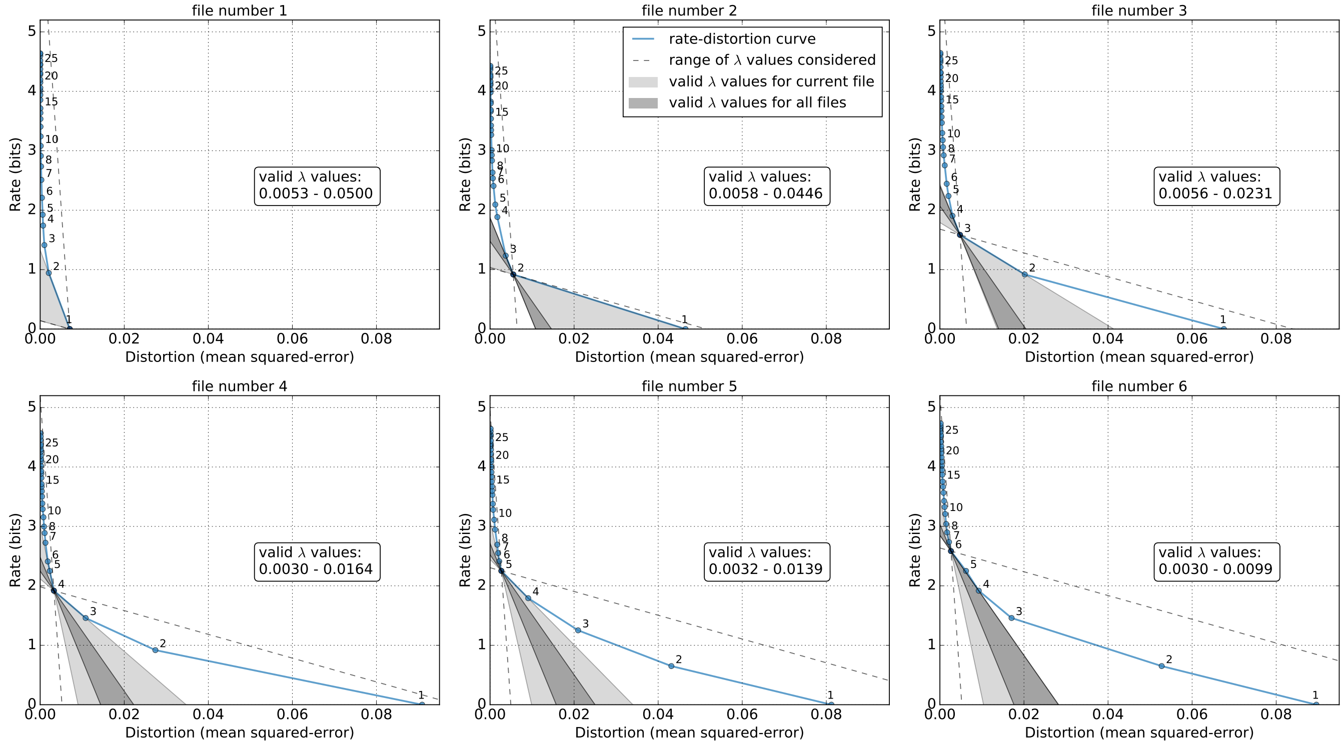


Fig. 8. Rate-distortion curves (blue) for the synthetic audio files. Extremes of the grid of λ values are depicted with dashed lines. The range of valid λ values for each file and the range in common for all files are shown as light and dark grayed out regions, respectively.

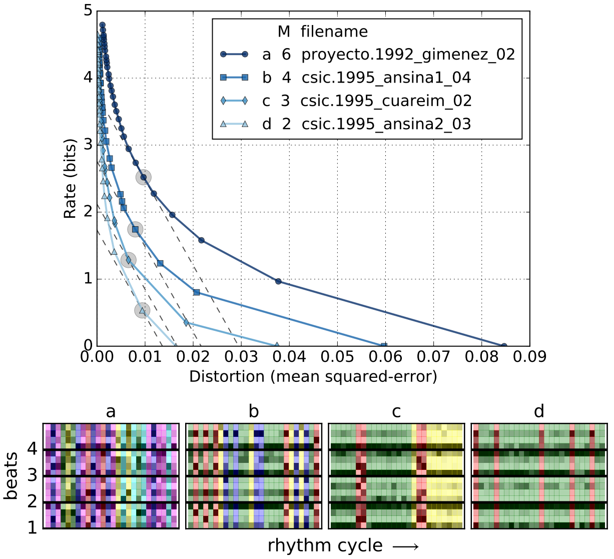


Fig. 9. Estimation of the number of rhythmic patterns for the performances of Fig. 5. Dashed lines intersecting each rate-distortion curve (top) represent the discrete Lagrangian minimization applied. The resulting coding using the estimated number of patterns as the codebook size is depicted with colors (bottom).

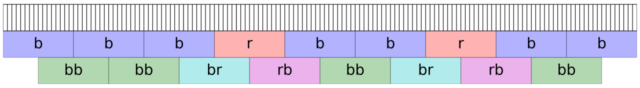


Fig. 10. Schematic representation of two possible rhythmic pattern alignments which imply different codebook sizes.

cation of beats is known, the aim is to identify which one of them corresponds to the downbeat. This is addressed by considering the different possible alignments of the four beats within the rhythm cycle. When rhythmic patterns of one-cycle length are considered, their alternation along the whole performance can give a hint on the location of the downbeat. In particular, the correct alignment will probably allow for a less complex description of the input sequence when compared to the misaligned options, since the latter will likely require more clusters to accommodate the patterns variability caused by the misalignment.

This is schematically illustrated in the example of Fig. 10. Consider the input vectors $\mathrm{x}_j, j \in [1,N]$, one after the other as a single stream of features. To produce the input sequence x^N, they have to be assembled in groups with the length of a rhythm cycle, which is 16 *tatum* pulses in this case. Suppose there are only two different rhythmic patterns played, say a *base* and a *repicado* pattern (notated as b and r in Fig. 10). Therefore, the correct alignment—the one consistent with the downbeat—can be optimally represented with a codebook of only two codevectors. However, other alignments will produce rhythmic patterns that are combinations of the original ones, yielding *base-repicado* (br), *repicado-base* (rb) and *base-base* (bb) patterns. Thus, a codebook of three codevectors is needed, leading to a more complex description of the input sequence.

When the downbeat detection of an audio file is handled, each beat of the four-beat rhythm cycle is alternatively considered as the downbeat, so four different alignments have to be evaluated. The different alignments are implemented as circular shifts of the feature map, starting from a shift of

0 beat (i.e. no shift) up to a shift of 3 beats. Larger shifts are redundant and therefore not considered—e.g. a shift of 4 beats produces no shift. Then, operational rate-distortion curves are computed for each different alignment.

This is shown in Fig. 11 for three of the synthetic audio files of Section 3.3, involving 3, 4 and 5 rhythmic patterns respectively. The complexity measures are also included in Fig. 11 for each one of the shifts, namely the area under curve (AUC), and the minimum value of the Lagrangian rate-distortion functional (Jmin). The discrete Lagrangian method is applied using a value of $\lambda^* = 0.00785$ (the mean of the range $\lambda_{[1,6]}$), and the codebook size M obtained in this way is also indicated. It can be seen that for all the audio files the correct alignment produces a curve that takes lower rate-distortion values compared to the shifted ones. Note that complexity measures also show this behavior, and that even in those cases when the codebook size is the same the correct alignment yields a smaller distortion value, which indicates that each group of patterns is better represented by a single codevector.

The previous examples indicate that the downbeat could be identified by comparing the rate-distortion trade-off of the different alignments. Nevertheless, the rationale for this is the alternation of patterns in the recording, and there may be some cases which fail to provide enough information for downbeat detection. To further illustrate this, the analysis of two real recordings of the dataset, namely e and d, is considered in the following.

Recall the diagram of Fig. 10, in which the correct alignment yields the shortest codebook size. This situation is illustrated in Fig. 12, for recording e, which contains *base* and *repicado* patterns. The shifting of the feature map gives rise to a higher number of rhythmic patterns, so the complexity of the description needed to account for the performance is increased. Note that in this case both complexity measures favor the selection of the correct alignment.

However, it is fairly obvious that if the performance contains a single pattern all alignments will be equivalent. Moreover, even in the case where there is more than one pattern the complexity of the different alignments may look all the same. For instance, in recording d—the most simple of the recordings introduced in Fig. 5—the differences between the patterns are confined to a single beat, as shown in Fig. 13. There is only one *base* pattern throughout the whole performance which sometimes shows an ornamentation in the fourth beat. Thus, shifting the patterns only relocates the ornamentation to a different beat, so the rate-distortion curves and measures provide no evidence to prefer one alignment over the others.

This second example stresses the fact that the proposed downbeat detection approach requires not only the alternation of different rhythmic patterns but also that the differences between them span over the whole rhythmic cycle, as in recording e (see Fig. 12). In fact, if there were no differences between the rhythmic patterns for a certain beat during the entire recording, i.e. the four *tatums* of the beat always were articulated in the same way, then this beat would carry no information regarding the location of the downbeat and ambiguity would arise between different shifts. Nevertheless, note that differences between *base* and *repicado* patterns usually extend over the whole rhythmic cycle (see for instance the music scores of Fig. 7). Consequently, the method is likely to succeed for a performance that alternates the typical *base* and *repicado* patterns. On the other hand, if the performance is too simple, the downbeat may not be identified correctly. It is interesting to note that the degree of complexity of the performance could be estimated beforehand, even without knowing the location of the downbeat. The AUC or Jmin value for an arbitrary alignment could be used for that purpose, since their values are very similar for the different alignments. More reliable measures could be devised to assess the degree of confidence in the downbeat estimation taking into account the extent of the differences between the rhythmic patterns.

3.4.1 Experiment 5

The proposed approach for downbeat detection is tested herein over the whole dataset. For each recording the low-frequency feature was extracted and the beat/downbeat labels were used to render the four different alignments of the beats within the rhythm cycle. Then, for each different alignment the operational rate-distortion curve was computed and the complexity measures were calculated. The downbeat was estimated as the beat corresponding to the shift that minimizes the complexity measure. A process similar to that of Fig. 8 was applied in a leave-one-out scheme for determining the λ value for the discrete Lagrangian method. The overall correct detection attained is 65.7% for the AUC, and 74.3% for the Jmin measure. [3]

As expected, some recordings are troublesome for the method, such as the recording analyzed in Fig. 13 and the three other recordings of the same performer (the AUC measure criterion fails in all of them, while the Jmin measure misses two). Apart from having the lowest degree of complexity of the whole dataset (i.e. AUC value), all of them consist of a single pattern occasionally ornamented in the fourth beat, and as previously noted fail to provide enough information for downbeat detection. Both measures fail in some other recordings, which exhibits only a single *base* pattern with a few simple variations. In this case, the patterns show virtually no differences at a certain beat during the entire performance, thus leading to ambiguity in the selection of the downbeat. It is interesting to note that for a large number of the recordings (22/35, 62.9%) the estimation of the downbeat is correct with both measures.

4 Discussion and conclusions

In this paper a novel approach for percussion music analysis based on information theory concepts was proposed. Given an audio recording of a percussion music performance, one computes a lossy representation that captures much of its underlying regularity but tolerates some

[3] The values of the complexity measures obtained for each file of the dataset are provided in the companion web page `https://iie.fing.edu.uy/~rocamora/JAES2018/`.

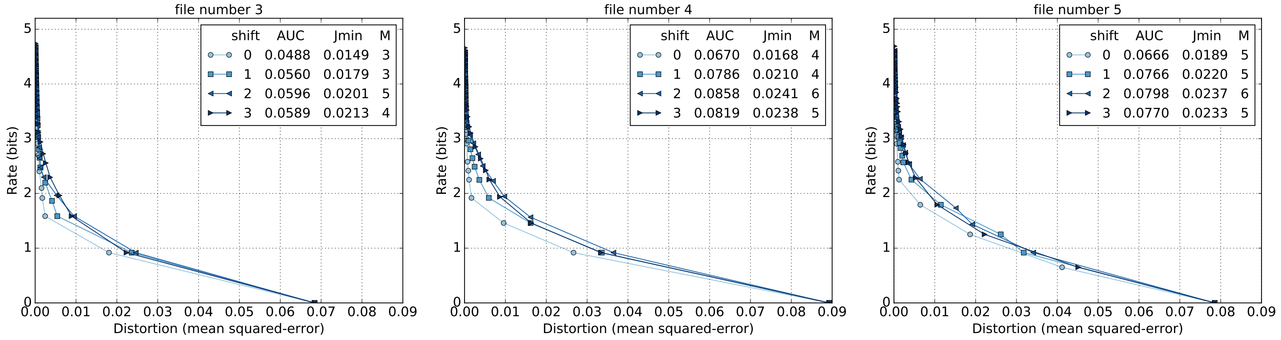


Fig. 11. Downbeat detection analysis for three of the synthetic audio files introduced in Figure 7, involving 3, 4 and 5 rhythmic patterns. The rate-distortion curves correspond to the four different possible alignments of the beats within the rhythm cycle.

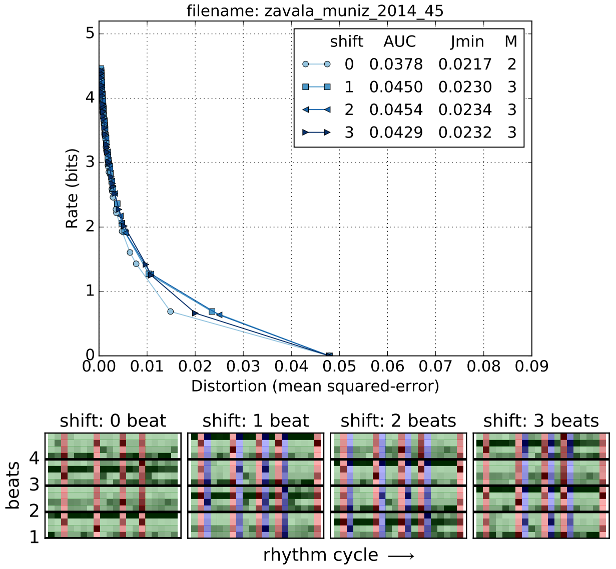


Fig. 12. Downbeat detection analysis for a recording of the dataset (e) with *base* and *repicado* patterns. The rate-distortion curves and the feature maps excerpts correspond to the four different possible alignments of the beats within the rhythm cycle.

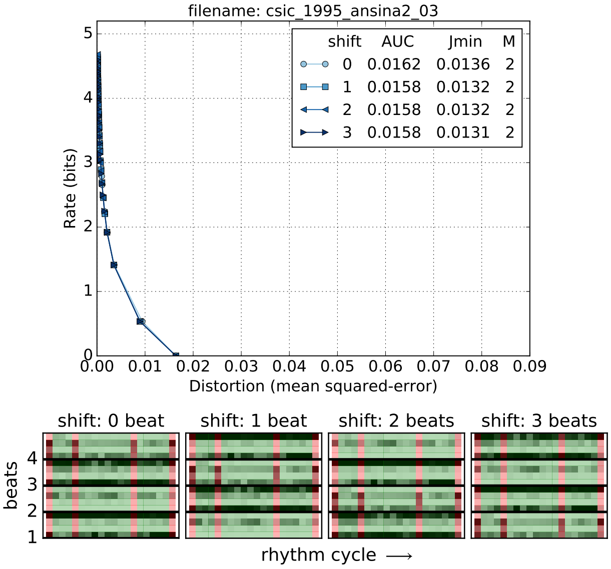


Fig. 13. Downbeat detection analysis for a recording of the dataset (d) with only a *base* pattern occasionally ornamented in the fourth beat. The four different possible alignments are considered.

amount of distortion. Thus, within a rate-distortion theory framework, the study of the trade-off between rate and distortion allows for the extraction of some relevant information about the performance.

Several experiments were conducted in order to assess the usefulness of the proposed approach when applied to a dataset of *candombe* drumming audio recordings. In particular, different performances were compared according to a measure of their overall complexity drawn from the operational rate-distortion curve, yielding results which roughly correspond to subjective judgment and correlate well with personal style and expertise. In addition, the estimation of the number of different rhythmic patterns in the recording was posed as the problem of selecting an operational point of the rate-distortion curve. The outcome of this method provided compact representations of the performances that are quite in accordance with manual analysis. Finally, the downbeat detection task from an audio signal was formulated as a data compression problem. To do that, the different possible alignments of the beats within the rhythm cycle were considered, and the one providing the most succinct representation—in terms of the rate-distortion trade-off—was selected as the downbeat. The method proved to be effective for a large part of the dataset, and the underlying rationale for its success as well as its limitations were discussed and illustrated with examples.

Previous work reported better downbeat detection results on the same dataset [34].[4] However, the previous approach is based on tracking rhythmic patterns that are informed to the algorithm, either based on a priori musical knowledge about the rhythm, or learned from the labeled database. The herein proposed method constitutes a novel idea for tackling the downbeat detection problem that is less grounded on high-level information about the rhythm

[4] Result for the Jmin measure corresponds to an Fmeasure of 74.3, whereas values between 76.9 and 80.6 are reported in [34].

or in a training scheme, and could be combined with any other existing method as another source of information.

A natural extension of the present work is taking into account the cost of describing the chosen model itself, as in the Minimum Description Length (MDL) approach. Besides, other information theory frameworks for model selection and the application of the proposed approach to other types of music that exhibit repeated rhythmic patterns (e.g. Afro-Brazilian) will be tackled in future work.

5 ACKNOWLEDGMENT

The authors thank the performers and all people involved in creating the dataset; the music expert Prof. L. Jure for his invaluable support and advice; Dr. F. Lecumberry and Dr. I. Ramírez for fruitful discussions on information theory. This work was partly supported by funding agencies CAPES and CNPq (Brazil), ANII and CSIC (Uruguay).

6 REFERENCES

[1] M. J. Kearns, U. V. Vazirani, *An Introduction to Computational Learning Theory* (MIT, USA) (1994), doi:10.7551/mitpress/3897.001.0001.

[2] A. Lempel, J. Ziv, "On the complexity of finite sequences," *IEEE Trans. on Information Theory*, vol. 22, no. 1, pp. 75–81 (1976), doi:10.1109/TIT.1976.1055501.

[3] J. Ziv, A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. on Information Theory*, vol. 23, no. 3, pp. 337–343 (1977), doi:10.1109/TIT.1977.1055714.

[4] P. Juola, "Assessing Linguistic Complexity," in M. Miestamo, K. Sinnemäki, F. Karlsson (Eds.), *Language Complexity: Typology, Contact, Change*, chap. 6, pp. 89–108 (John Benjamins, Netherlands) (2007).

[5] J. Rigau, M. Feixas, M. Sbert, "An information-theoretic framework for image complexity," presented at the *1st. Eurographics Conf. on Computational Aesthetics in Graphics, Visualization and Imaging (CompAesth 05)*, pp. 177–184 (2005).

[6] M. Aboy, R. Hornero, D. Abasolo, D. Alvarez, "Interpretation of the Lempel-Ziv complexity measure in the context of biomedical signal analysis," *IEEE Trans. on Biomedical Engineering*, vol. 53, no. 11, pp. 2282–2288 (2006), doi:10.1109/TBME.2006.883696.

[7] M. M. Marin, H. Leder, "Examining complexity across domains: Relating subjective and objective measures of affective environmental scenes, paintings and music," *PLoS ONE*, vol. 8, no. 8, p. e72412 (2013), doi:10.1371/journal.pone.0072412.

[8] T. Eerola, "Expectancy-violation and information-theoretic models of melodic complexity," *Empirical Musicology Review*, vol. 11, no. 1 (2016), doi:10.18061/emr.v11i1.4836.

[9] S. Streich, *Music Complexity: A multi-faceted description of audio content*, Ph.D. thesis, Dept. of Technology, Universitat Pompeu Fabra (2007).

[10] M. Müller, "Music Structure Analysis," in *Fundamentals of Music Processing*, chap. 4, pp. 167–236 (Springer, Switzerland) (2015), doi:10.1007/978-3-319-21945-5_4.

[11] B. Janssen, W. B. de Haas, A. Volk, P. van Kranenburg, "Finding repeated patterns in music: State of knowledge, challenges, perspectives," presented at the *10th Int. Symp. on Computer Music Multidisciplinary Research (CMMR 2013)*, pp. 225–240 (2013).

[12] J. L. Hutchens, M. D. Alder, "Finding structure via compression," presented at the *Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning (NeMLaP3/CoNLL '98)*, pp. 79–82 (1998), doi:10.3115/1603899.1603913.

[13] C. Louboutin, D. Meredith, "Using general-purpose compression algorithms for music analysis," *Journal of New Music Research*, vol. 45, no. 1, pp. 1–16 (2016), doi:10.1080/09298215.2015.1133656.

[14] D. Meredith, "Analysing Music with Point-Set Compression Algorithms," in D. Meredith (Ed.), *Computational Music Analysis*, pp. 335–366 (Springer, Switzerland) (2016), doi:10.1007/978-3-319-25931-4_13.

[15] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656 (1948), doi:10.1002/j.1538-7305.1948.tb01338.x.

[16] C. E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," in *IRE National Convention Record, Part 4*, pp. 142–163 (1959).

[17] T. Berger, "Rate Distortion Theory and Data Compression," in International Centre for Mechanical Sciences (Ed.), *Advances in Source Coding*, vol. 166, chap. 1, pp. 3–39 (Springer, USA) (1975), doi:10.1007/978-3-7091-2928-9_1.

[18] T. Berger, J. D. Gibson, "Lossy source coding," *IEEE Trans. on Information Theory*, vol. 44, no. 6, pp. 2693–2723 (1998), doi:10.1109/18.720552.

[19] A. Ortega, K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 23–50 (1998), doi:10.1109/79.733495.

[20] M. Schedl, E. Gómez, J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations and Trends in Information Retrieval*, vol. 8, no. 2-3, pp. 127–261 (2014), doi:10.1561/1500000042.

[21] M. Müller, D. P. W. Ellis, A. Klapuri, G. Richard, "Signal processing for music analysis," *IEEE J. of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110 (2011), doi:10.1109/JSTSP.2011.2112333.

[22] B. Di Giorgi, M. Zanoni, S. Böck, A. Sarti, "Multipath Beat Tracking," *J. Audio Eng. Soc*, vol. 64, no. 7/8, pp. 493–502 (2016), doi:10.17743/jaes.2016.0025.

[23] D. Gärtner, "Unsupervised Learning of the Downbeat in Drum Patterns," presented at the *AES 53rd International Conference on Semantic Audio* (2014).

[24] J. London, *Hearing in Time: Psychological Aspects of Musical Meter* (Oxford, USA) (2004).

[25] F. Lerdahl, R. Jackendoff, *A Generative Theory of Tonal Music* (MIT, USA) (1985).

[26] T. M. Cover, J. A. Thomas, *Elements of Information Theory* (Wiley, USA), 2nd ed. (2006).

[27] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137 (1982), doi:10.1109/TIT.1982.1056489.

[28] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 28, no. 1, pp. 84–95 (1980), doi:10.1109/TCOM.1980.1094577.

[29] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666 (2010), doi:10.1016/j.patrec.2009.09.011.

[30] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, no. 159 in The Kluwer International Series in Engineering and Computer Science (Kluwer, New York, USA) (1992).

[31] G. Andrews, *Blackness in the White Nation: A History of Afro-Uruguay* (UNC, Chapel Hill, USA) (2010), doi:10.5149/9780807899601_andrews.

[32] S. Dixon, "Onset Detection Revisited," presented at the *9th Int. Conf. on Digital Audio Effects*, pp. 133–137 (2006).

[33] H. Fastl, E. Zwicker, *Psychoacoustics: Facts and Models* (Springer, Germany), 3rd ed. (2006).

[34] L. Nunes, M. Rocamora, L. Jure, L. W. P. Biscainho, "Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan candombe drumming," presented at the *16th Int. Society for Music Information Retrieval Conf. (ISMIR 2015)*, pp. 264–270 (2015).

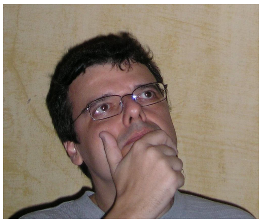
[35] M. Rocamora, L. Jure, B. Marenco, M. Fuentes, F. Lanzaro, A. Gómez, "An audio-visual database of candombe of performances for computational musicological studies," presented at the *II Congreso Intl. de Ciencia y Tecnología Musical (CICTeM 2015)*, pp. 17–24 (2015).

[36] M. Rocamora, L. Jure, L. W. P. Biscainho, "Tools for detection and classification of piano drum patterns from candombe recordings," presented at the *9th Conf. on Interdisciplinary Musicology (CIM 2014)*, pp. 382–387 (2014).

THE AUTHORS

Martín Rocamora

Pablo Cancela



Luiz W. P. Biscainho

Martín Rocamora worked in the telecommunication industry between 1998 and 2005. He received the B.Sc, M.Sc. and D.Sc. degrees in Electrical Engineering in 2004, 2011 and 2018 respectively, from the School of Engineering at Universidad de la República, Uruguay. In 2005 he started working as lecturer at the same university, where he currently holds a position as Assistant Professor in Signal Processing. His research interests include music information retrieval, computational musicology, digital audio signal processing and machine learning. He is currently a member of the IEEE (Institute of Electrical and Electronics Engineers) and the AES (Audio Engineering Society).

●

Pablo Cancela received the Electrical Engineering and Computer Science degree from Faculty of Engineering (FI), Universidad de la República (Udelar), Uruguay in 2002. He received his PhD in Electrical Engineering also from FI in 2016. He works at the Electrical Engineering Department of FI, Udelar since 2001, currently as an Adjunct Professor. His main research interests include signal processing and machine learning.

●

Luiz Biscainho was born in Rio de Janeiro, Brazil, in 1962. He received the Electronics Engineering degree (magna cum laude) from the EE (now Poli) at Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1985, and the M.Sc. and D.Sc. degrees in Electrical Engineering from the COPPE at UFRJ in 1990 and 2000, respectively. Having worked in the telecommunication industry between 1985 and 1993, Dr. Biscainho is now Associate Professor at the Department of Electronics and Computer Engineering (DEL) of Poli and the Electrical Engineering Program (PEE) of COPPE at UFRJ. His research area is digital audio processing. He is currently a member of the IEEE (Institute of Electrical and Electronics Engineers), the AES (Audio Engineering Society), the SBrT (Brazilian Telecommunications Society), and the SBC (Brazilian Computer Society).