UNCOVERING AUDIO PATTERNS IN MUSIC WITH NONNEGATIVE TUCKER DECOMPOSITION FOR STRUCTURAL SEGMENTATION

Axel Marmoret[1] **Jérémy E. Cohen**[1] **Nancy Bertin**[1] **Frédéric Bimbot**[1]
[1]Univ Rennes, Inria, CNRS, IRISA, France.
axel.marmoret@irisa.fr

ABSTRACT

Recent work has proposed the use of tensor decomposition to model repetitions and to separate tracks in loop-based electronic music. The present work investigates further on the ability of Nonnegative Tucker Decompositon (NTD) to uncover musical patterns and structure in pop songs in their audio form. Exploiting the fact that NTD tends to express the content of bars as linear combinations of a few patterns, we illustrate the ability of the decomposition to capture and single out repeated motifs in the corresponding compressed space, which can be interpreted from a musical viewpoint. The resulting features also turn out to be efficient for structural segmentation, leading to experimental results on the RWC Pop data set which are potentially challenging state-of-the-art approaches that rely on extensive example-based learning schemes.

1. INTRODUCTION

A common problem in Music Information Retrieval domain (MIR) is the design of musical content representations and features able to capture meaningful information in relation to a particular aspect of music. While short-term features are dominant in the literature, higher-scale features aiming to describe medium-term patterns and long-term structural properties tend to be much less addressed.

Recent work by Smith and Goto [1] has proposed the use of tensor decomposition to model repetitions in loop-based electronic music, with the purpose of separating tracks in audio content. In this paper, we explore the ability of the method to provide a sparse description of music by capturing and characterizing patterns at the bar-scale level in western pop songs in their audio form. As a testbed, we evaluate the effectiveness of the new features for structural segmentation, *i.e.* the task of retrieving the boundaries of the various musical sections (such as verses, choruses, intros, bridges...) which form a music piece.

We first recall (in section 2) the mathematical theory of the tensorial model called Nonnegative Tucker Decomposition (NTD), and we provide detailed illustrations and interpretation of the NTD components on the audio recording of a well-known pop song. We then (in section 3) elaborate on a number of practical considerations related to NTD, which are needed to be taken into account when applying the model to real music data. In the last part of the article (sections 4 and 5), we report on experiments and results obtained with the NTD-derived features for structural segmentation of the RWC Pop Music data set [2].

2. NONNEGATIVE TUCKER DECOMPOSITION

2.1 Time-Frequency-Bar Tensor

Music in its audio form is often represented in the time-frequency domain as a spectrogram, *i.e.* a 2-dimensional matrix (further denoted as X). Along the x-axis, the temporal dimension unfolds, discretized as signal frames, while the y-axis is a frequency-related dimension (such as modules of the Fourier coefficients, pitches, constant-Q transforms, wavelet coefficients...).

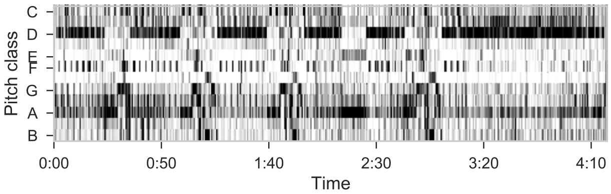
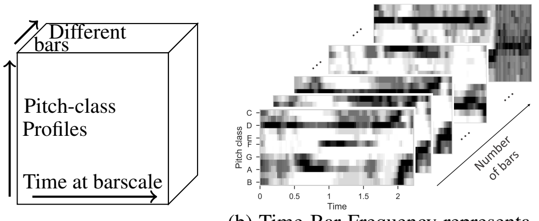


Figure 1: Chromagram of “Come Together”, by The Beatles.

In this work, we describe music as chromagrams. Conventionally, the 12 rows represent the energy distribution across the song for each semi-tone of the classical western music scale, where a note and all its octave counterparts are represented in the same row. An example chromagram is shown in Figure 1.

In the tensorial approach, the temporal dimension is broken up into two distinct dimensions: a low-scale dimension representing time in terms of frame index normalized at the bar scale, and a high-scale time dimension representing the bar index within the entire piece. This new viewpoint makes it possible to represent a song as a third-order tensor $\mathcal{X}$ of size $F \times T \times B$, F being the size of the frequency dimension (12 in the case of chromas), T the number of frames used to describe bars (local time scale) and B the number of bars in the song (global time scale). We call $\mathcal{X}$ the Time-Frequency-Bar representation of the song which, from a data structure viewpoint, is the recasting of X as a 3D array. Tensor $\mathcal{X}$ can be seen as the concatenation

Attribution: Axel Marmoret, Jérémy E. Cohen, Nancy Bertin, Frédéric Bimbot, “Uncovering audio patterns in music with Nonnegative Tucker Decomposition for structural segmentation”, in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.



(a) Tensor: multi-dimensional array

(b) Time-Bar-Frequency representation of "Come Together" as a concatenation of barwise chromagrams

Figure 2: Principle and illustration of a Time-Bar-Frequency representation as a third-order tensor.

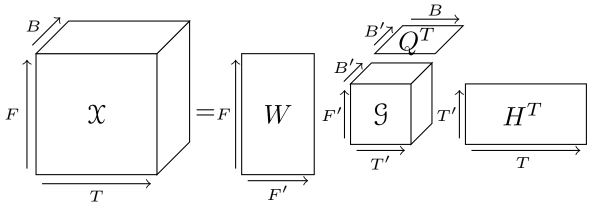


Figure 3: Nonnegative Tucker Decomposition of tensor $\mathcal{X}$ in factor matrices W, H, Q, and core tensor $\mathcal{G}$, with their dimensions.

of local time-frequency representations, each of them characterizing the content of a bar, as illustrated on Figure 2. Note that, as bars can be of different lengths in absolute time, the frame hop depends on each bar, and is defined so that all bars contain the same number of frames.

2.2 Mathematical Model and Formalism

Let us denote as $\mathcal{X}$ the Time-Frequency-Bar representation of a song, with dimensions $F \times T \times B$. Assuming chroma coefficients are all nonnegative, $\mathcal{X}$ is also a nonnegative tensor. Computing a Nonnegative Tucker Decomposition (NTD) of $\mathcal{X}$ consists in finding 3 nonnegative factor matrices W, H and Q (corresponding to the three "modes" of the Time-Frequency-Bar tensor) and a nonnegative core tensor $\mathcal{G}$ which relates the three modes as of how to combine them to reconstruct (an approximation of) $\mathcal{X}$.

The dimensions $F' \times T' \times B'$ of the core tensor $\mathcal{G}$ are usually set to be lower than those of $\mathcal{X}$ (*i.e.* $F', T', B' \leq F, T, B$ respectively). As a consequence, matrices W, H and Q are respectively of dimensions $F \times F'$, $T \times T'$ and $B \times B'$ and they can be understood as transformed and compressed representations of the raw information conveyed across the three dimensions of the full tensor.

In conventional tensor-product notation [3], the approximation of $\mathcal{X}$ can be written in compact form as:

$$\mathcal{X} \approx \mathcal{G} \times_1 W \times_2 H \times_3 Q\ . \tag{1}$$

which rewrites, using element-wise notation, as:

$$\mathcal{X}(f,t,b) \approx \sum_{f',t',b'=1}^{F',T',B'} \mathcal{G}(f',t',b')W(f,f')H(t,t')Q(b,b') \tag{2}$$

In particular, any given bar of index b is represented as:

$$\mathcal{X}(:,:,b) \approx W \left(\sum_{b'=1}^{B'} Q(b,b')\mathcal{G}(:,:,b') \right) H^T \tag{3}$$

Figure 3 depicts a schematic 3-D representation of a NTD. NTD core dimensions F', T' and B' are assumed to be known (or set empirically) prior to the decomposition. As they are lower than the dimensions of their respective mode of the tensor, NTD achieves information compression via nonlinear dimensionality reduction. Indeed, for an original tensor of size $F \times T \times B$ (*i.e.* comprising $F.T.B$ numerical values), the NTD decomposition will total $F.F' + T.T' + B.B' + F'.T'.B'$ values. For example, in the decomposition presented further in Figure 4, the original tensor contains 102528 real positive values ($F, T, B = 12, 96, 89$), while only 3626 for the NTD ($F', T', B' = 12, 12, 10$).

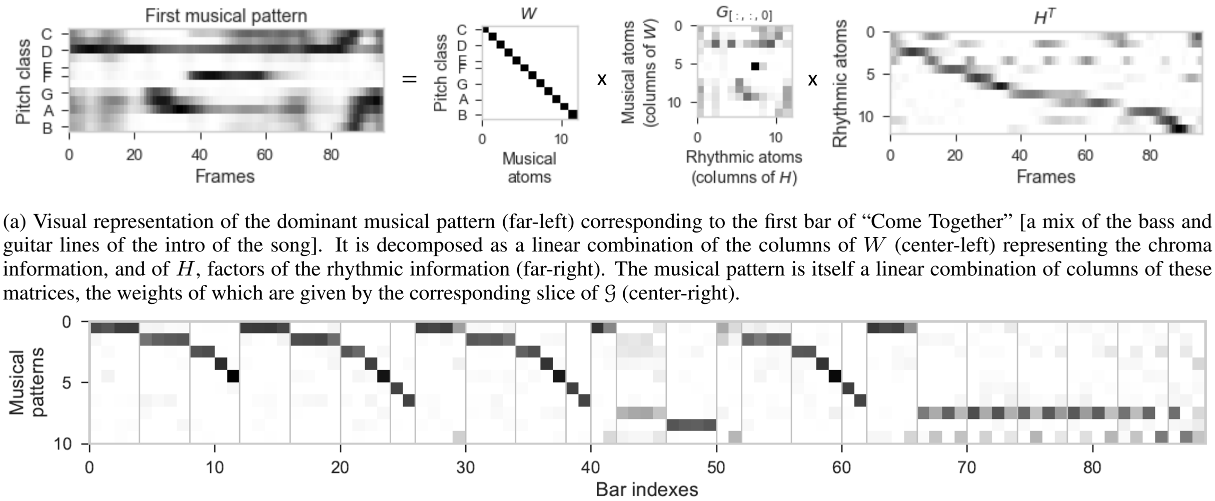
2.3 Interpretation of the NTD

Loosely speaking, music can be viewed as composed of musical events (notes, percussive sounds, ...) occurring non-randomly in bars. Under that assumption, bars can be modeled as the combination of a limited set of time-frequency templates along time within a bar, according to some rhythmic values, such as "half notes" or "beamed eight notes" for example. This is the purpose of a conventional musical score, where the major part of symbolic information represents pitch and rhythm. Following that idea, it is a very popular goal in MIR to design methods for turning back musical content (in audio form) into a sparse combination of musical events and temporal activations, as is the case, for instance, with Nonnegative Matrix Factorization (NMF) for music transcription [4]. Moreover, music often contains repetitions: different bars can entirely or partly share similar content. For instance, beside almost identical repetitions, some instrumental lines can reoccur in different contexts: an identical bass line in a verse and in a guitar solo, for example.

Combining these observations, we assume that each bar can be represented as the nonnegative combination of a few "musical patterns" (as NMF would do), where a "musical pattern" is itself a sparse combination of musical events and rhythmic activations at the bar scale (for example a melodic line, or a drum fill). Repetitions imply that some of these musical patterns should appear in several bars across a piece. NTD offers an ideal framework to model these properties for music decomposition, musical patterns being efficiently and sparsely shared across bars.

In the NTD, the W matrix represents the musical events, such as the most recurrent notes or chords. H represents rhythmic activations at the bar scale, for example 4 quarter notes on the beats. Then, each 2D "slice" of the core $\mathcal{G}$ linking these two matrices defines a musical pattern, as a linear combination of some of their columns (musical and rhythmic atoms): for example bass drum hits on the on-beats and snare hits on the off-beats. Finally, Q indicates, for each bar, the combination of musical patterns forming it (generally a few) and their respective intensity.

Figure 4 provides a detailed example of the various NTD components stemming from "Come Together" by the Beatles. While the upper part of the figure illustrates the



(a) Visual representation of the dominant musical pattern (far-left) corresponding to the first bar of "Come Together" [a mix of the bass and guitar lines of the intro of the song]. It is decomposed as a linear combination of the columns of W (center-left) representing the chroma information, and of H, factors of the rhythmic information (far-right). The musical pattern is itself a linear combination of columns of these matrices, the weights of which are given by the corresponding slice of $\mathcal{G}$ (center-right).

(b) "Come Together" represented as its Q^T matrix. Each row represents a musical pattern, in their order of appearance in the piece. Grey lines represent the segmentation annotation.

Figure 4: Visualizations of the NTD of Come Together by The Beatles, with ranks $T' = 12$ and $B' = 10$.

dominant musical pattern for the first bar together with its decomposition in NTD, the lower part depicts the description of the entire piece via the Q^T matrix of the NTD. This example has been obtained from the chromagram represented in Figure 1. Because the song is expressed on the 12-chroma scale, we expect little compressibility with respect to this dimension. We hence simplify the model by fixing W to the 12-size identity matrix. This means that each semi-tone is represented by one and only one column of W. For higher dimensions or different representations, columns of W could represent a wider range of harmonic or percussive sounds, chords, or any other frequency pattern. Conversely, ranks T' and B' (respectively the second dimension for H and Q) are adjustable parameters of the model. In the decomposition presented in Figure 4, they have been set to $T' = 12$ and $B' = 10$. All columns of H and all slices of the core linking the factor matrices W and H, which define the musical patterns, are l_2 normalized (*i.e.* divided by their standard deviation).

3. PRACTICAL INSIGHTS ON THE NTD

In this section, we discuss a number of considerations which are bound to have an impact on the actual result of the NTD-based representations and must therefore be taken into account in practical situations.

3.1 NTD Algorithm

NTD can typically be computed by minimizing the following non-convex objective function with respect to the nonnegative matrices W, H, Q and the core tensor $\mathcal{G}$:

$$\|\mathcal{X} - \mathcal{G} \times_1 W \times_2 H \times_3 Q\|_F^2 \tag{4}$$

While a direct global minimization of Eqn (4) is not tractable in general, a standard approach in the literature is to resort to alternating optimization. Following [5], we solve Eqn (4) for W, H, Q and $\mathcal{G}$ alternatively. It can be shown that each of these steps means solving a matrix nonnegative least-square problem of the form:

$$\min_{Z \geq 0} \|Y - AZ\|_F^2 \tag{5}$$

for some matrices Y, A, Z. This problem is convex, and it is possible to solve it exactly, or up to an arbitrary precision. An efficient algorithm for solving matrix nonnegative least squares with high precision is the Hierarchical Alternating Least Squares, and we used an accelerated variant of it to speed-up computation [6]. The problem of updating $\mathcal{G}$ is also a nonnegative least squares problem, but not a matrix one. Therefore, to update the core tensor $\mathcal{G}$, we used a proximal gradient with optimal step [7, Ch. 10].

It can be shown that the proposed alternating algorithm is guaranteed to converge to a stationary point of the objective function (4), since it boils down to an alternating proximal gradient algorithm with optimal step [8]. In practice, we used a stopping criterion based either on a fixed maximal number of iterations or on a fixed minimal tolerance of improvement between two successive updates. The entire code, along with experimental notebooks, are published and open-source [1]. Under this implementation, computing the NTD for "Come Together" (4'16" song) with our algorithm takes approximately 15 seconds on a laptop with an Intel® Core(TM) i7 processor and 16GB of RAM.

3.2 Robustness of the NTD

At least two issues with the NTD make the output of any algorithm highly dependent on the initialization. First, there might be several solutions $W, H, Q, \mathcal{G}$ that provide the same (or a very similar) estimate $\hat{\mathcal{X}} \approx \mathcal{X}$. This problem, known as identifiability deficiency, has been little studied for NTD, and established identifiability conditions are very restrictive [9]. Moreover, these conditions are hard to check in practice. Therefore it is unreasonable to assess the identifiability of the NTD in our application. As a consequence, this means that there might be infinitely many

[1] https://gitlab.inria.fr/amarmore/musicntd/-/tree/0.1.0

solutions to minimizing Eqn (4) that are, from an optimization point of view, equally satisfying. Second, even in the case where the NTD is identifiable, the cost function of (4) is highly non-convex, and local algorithms can only hope to recover a local minimum at best.

These two issues combined give rise to a high dependency of the solution on the initial condition: from two different initializations, two different results – most probably non-identifiable local minima – are likely to be obtained. We have observed such situations in our investigations, with various initializations indeed resulting in different outputs. However, in most cases the decomposition would provide results that were reasonably interpretable from a musical perspective. In particular, when we initialized the algorithm with the absolute values of the Higher Order SVD [10] computed with the Tensorly toolbox [11], the procedure consistently provided satisfying results for segmentation, as detailed further in our experiments.

3.3 Rank Selection

The ranks F', T' and B' of the decomposition are crucial parameters of the NTD model. Indeed, low ranks tend to over-compress information in the data, failing to uncover relevant structural information in the song, while high ranks may give too much importance to details in the data, resulting in the unability of the model to group similar patterns in a same class of representations.

As developed further in section 5.4, our experiments indicate that the optimal ranks are probably specific for each song, which can be easily understood as a consequence of the diversity of intrinsic variability across music pieces. Providing an efficient method for selecting the ranks is a challenging topic, left to future work.

4. NTD-BASED SEGMENTATION

To study further the relevance of the NTD representation, we evaluated it in the context of structural segmentation. To our knowledge, this is the first attempt to exploit tensorial representations for this purpose.

4.1 Autosimilarity for Describing Structure

The autosimilarity matrix X^TX of a music piece (X being its time-frequency representation) is commonly used in structural segmentation. Indeed, similar portions of the piece are likely to have high correlation values. A high density of high values around the diagonal is expected in passages with strong internal similarities, whereas low local correlations would indicate a change in homogeneity. In the ideal case, structural segments appear as consistent blocks with a high level of internal correlation while segment boundaries are points connecting such blocks, surrounded by zones of low cross-correlation.

Nonetheless, music signals usually generate dense autosimilarity matrices, as dissimilar segments in the musicological/perceptive sense (for example a guitar line on the chorus opposed to one in the verse) may still be close in terms of signal properties. While similar parts generate high correlation blocks, it can be harder to characterize segments boundaries when the same instruments are played in all segments (even when playing different lines).

In the present work, we replace X^TX by an autosimilarity matrix $\tilde{Q}\tilde{Q}^T$ computed from the row-wise normalized Q matrix (denoted $\tilde{Q}$), and study its capacity to provide an efficient representation for structural segmentation.

Our assumption is that bar descriptions provided by $\tilde{Q}$ provide a better contrast between similar and dissimilar musical constituents. For instance, we expect two different lines of the same instrument to generate different musical patterns, resulting in lower similarity, whereas compressive effects of NTD will increase correlation of similar events in the transformed space. In that sense, NTD can be seen as a way to uncover piece-dependent features for describing bars, which can then be used to group the bars according to their relative similarity.

Figure 5 depicts the “barwise” autosimilarity matrix of the chromagram X of “Come Together”: the content of each bar of the signal has been vectorized, and similarity is computed between these barscale vectors. This matrix is compared to the autosimilarity of the $\tilde{Q}$ matrix, presented on Figure 4b. This figure visually supports the hypothesis that autosimilarity matrices are sparser when computed from the matrix Q rather than from the chromas X. Still, highly similar blocks seem to be preserved.

4.2 A Segmentation Algorithm Using Autosimilarity

To assess this hypothesis, we implemented a segmentation algorithm based on the principle of a sliding convolution kernel along the diagonal of the autosimilarity matrix.

This kernel is a square binary matrix, whose entries are non-zero only on the lower and upper 4 subdiagonals around the main diagonal (Figure 6). In other terms, denoting k_{ij} the kernel elements, $k_{ij} = 1$ if $1 \leq |i-j| \leq 4$. Otherwise, $k_{ij} = 0$.

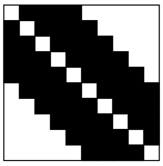


Figure 6: Kernel of size 10

For every possible segment (b_1, b_2), a kernel of size $n = b_2 - b_1 + 1$ is convolved with the corresponding autosimilarity sub-matrix (restricted to the bars between b_1 and b_2) which is then normalized by the size of the segment. This leads to a raw convolution score: $c_{b_1,b_2} = \frac{1}{n}\sum_{i,j=0}^{n-1} k_{ij}a_{i+b_1,j+b_1}$. The kernel aims at detecting local similarities within the 8 bars surrounding each bar. The more similar this surrounding is, the higher the score.

In addition, we combine the kernel score with a regularity penalty $p(n)$, depending on the size n of the segment. Indeed, in pop music in general (and in the MIREX10 RWC Pop annotations in particular [12]), the distribution of musical segment sizes (in bars) tend to be centered around 8, and they are more likely to be even than odd. In the experiments reported in the next section, we set empirically, $p(8) = 0$, $p(n) = \frac{1}{4}$, if n is a multiple of 4, $p(n) = \frac{1}{2}$ if n is a multiple of 2, and $p(n) = 1$ if n is odd. This penalty modifies the raw convolution score as

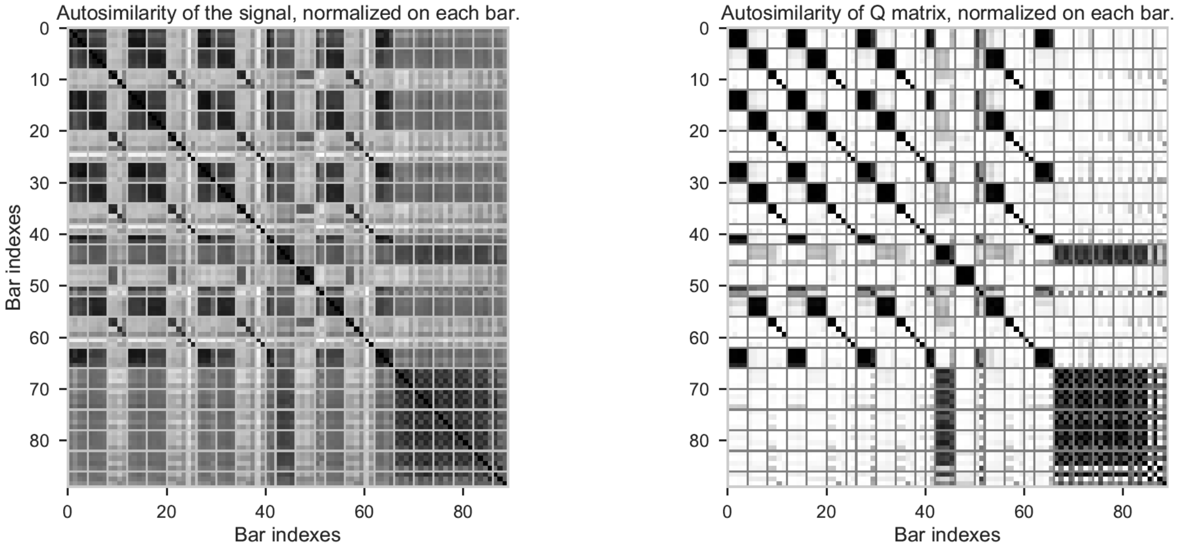


Figure 5: Barwise l_1-normalized autosimilarity matrices for "Come Together" (0: white - 1: black). Left: barwise chromagram autosimilarity - Right: autosimilarity of $\tilde{Q}$ matrix from Figure 4b. Grey horizontal and vertical lines represent the segmentation annotation.

follows:

$$c'_{b_1,b_2} = c_{b_1,b_2} - \lambda p(n) c_{k8}^{max} \qquad (6)$$

where c_{k8}^{max} is the maximum of the raw convolution score over all restrictions of size 8 bars within the piece, in order to cope with potential discrepancies in sparsity across autosimilarity matrices for different pieces. In practice, λ is fitted by cross-validation.

Finally, segment boundaries are found by a dynamic programming algorithm, inspired from [13]: it keeps the sequence of segments maximizing the global cost defined as the sum of all segment costs.

5. EXPERIMENTS

The proposed method was applied to the $\tilde{Q}^T$ representation and tested on the "structural segmentation" task, as defined in the MIREX campaigns [14], on the 100 songs from the RWC Pop database [2]. MIREX10 annotations [12] serve as the reference segmentation (1680 segments). We compare our results with state-of-the-art methods listed below.

5.1 Related Work

In the context of structural segmentation, numerous methods try to detect segment boundaries from the autosimilarity matrix, or from an "affinity matrix" derived from it.

The use of autosimilarity for segmenting music structure probably traces back to Foote [15]. In this work, structural boundaries are detected by applying a kernel along the diagonal, as described above. Foote's kernel though aims at detecting "novelty" in the signal's autosimilarity matrix, by comparing inter-similarity between the near past and near future at the current point. A high novelty should indicate a low inter-similarity between past and future, hinting towards a boundary between segments.

More recently, convex NMF was used for segmenting a pre-processed autosimilarity matrix [16]. A variant of NMF decomposition is used to enforce the feature space (here, similarity between different bars) to be contracted in convex combinations of columns of the autosimilarity matrix. Factorization results are thus interpreted as the most similar bars, which can then be processed into sections.

Spectral clustering can also be used. In [17], an affinity matrix is computed from the signal, where the similarity is obtained with k-nearest neighbors and time-proximity rules. Then, interpreting this matrix as a graph, and its values as vertices connectivity, this method studies the eigenvectors of its Laplacian. These eigenvectors can be interpreted as principally connected vertices, forming cluster classes for segmentation.

We primarily compare the NTD method with these techniques for two reasons. First, they are implemented in the MSAF toolbox [20]. Second, they reach state-of-the-art performance among "blind" methods for structural segmentation, *i.e.* methods which, like NTD, do not resort to extensive training from examples. Note that the segmentation results we obtained with MSAF, though, are slightly worse ($\approx 3/4\%$) than those obtained at MIREX 2016 [21], possibly due to evolutions of the toolbox itself in the interval. We did not tune any of the default parameters.

As current state-of-the-art, we selected the algorithm from [18] since it ranked first in this task in the last MIREX campaigns. However, as opposed to the previous methods, it requires supervised training from many examples.

5.2 Downbeat-Synchronous Alignment

By construction, the boundaries estimated by the NTD-based approach are aligned on downbeats, which is not the case for the techniques we use as baseline comparisons. As segments generally start and end on downbeats of the song, this alignment could induce a bias favouring our technique. To compensate for this, in addition to the segmentation scores computed with the original boundaries, we compute the scores after having aligned boundaries on the closest downbeat. We call this condition "Aligned on downbeats".

Algorithm		$P_{0.5}$	$R_{0.5}$	$F_{0.5}$	P_3	R_3	F_3
NTD-based autosimilarity		53.3%	62.1%	56.6%	66.8%	78.1%	71.1%
Barwise chromagram autosimilarity		43.1%	45.7%	43.9%	64.8%	68.0%	65.8%
Foote Novelty [15]	Original	29.7%	22.3%	25.1%	63.9%	48.6%	54.5%
	Aligned on downbeats	42.0%	30.0%	34.5%	67.1%	47.7%	55.0%
CNMF [16]	Original	22.8%	21.5%	21.5%	46.8%	45.1%	44.7%
	Aligned on downbeats	31.6%	28.1%	28.8%	50.7%	45.4%	46.5%
Spectral Clustering [17]	Original	31.2%	30.5%	29.4%	60.7%	60.8%	58.1%
	Aligned on downbeats	49.2%	45.0%	45.0%	65.5%	60.6%	60.3%

Table 1: Averaged segmentation scores, and their comparison with several “blind” reference methods.

Algorithm	$P_{0.5}$	$R_{0.5}$	$F_{0.5}$	P_3	R_3	F_3
NTD, with “oracle ranks” for each song	67.1%	78.2%	71.5%	78.5%	90.2%	83.1%
Neural Networks [18], results from MIREX 2015 [19]	80.4%	62.7%	69.7%	91.9%	71.1%	79.3%

Table 2: Averaged segmentation scores in the “oracle ranks” condition, compared to the current state-of-the-art (non-blind) method.

In addition, we also processed the barwise autosimilarity obtained directly from the chromagram, in order to measure the impact of the NTD-derived representation vs the raw time-frequency representation.

5.3 Implementation Details

RWC Pop signals are sampled at 44100Hz. Bars were estimated by the madmom toolbox [22]. Chromas were extracted from the Constant-Q Transform of the signal with 32-sample hop using Librosa [23], then mapped to 96 equally spaced chroma vectors per bar. This results in a chromagram X with 12 rows and $96 \times B$ columns. Tensors were handled with the Tensorly toolbox [11]. We use our own implementation of the NTD algorithm (see Section 3.1). Segmentation performance was computed with the mir_eval toolbox [24].

5.4 Results

Segmentation performance is evaluated with metrics based on “hit-rate”. The hit-rate considers a boundary as correct if it coincides with a boundary in the reference segmentation within some time window. From the count of correct and incorrect segment boundaries, we compute Precision, Recall and F-measure. Tolerance windows were chosen to be 0.5s and 3s, in line with MIREX standards.

As mentioned in Section 3.3, the ranks of the NTD strongly influence the decomposition and, consequently, the segmentation results. Ranks T' and B' are treated as adjustable parameters, and can vary between 12 to 48, with a step of 4. W is fixed to the 12-size identity matrix. The impact of T' and B' is investigated under two rank selection conditions.

In the first condition (Table 1), the RWC Pop data set is divided in two subsets (songs with odd vs even ID number), which are alternatively used as tuning (for global optimization of the ranks and the penalty parameter λ) and test data sets, in a 2-fold cross-validation fashion. Results shown in the table are averaged over the two folds. Hence, in this condition, all songs of a test data subset are decomposed with the same ranks, namely $T', B' = 40, 28$ for odd songs, and $48, 24$ for even ones.

In the second condition, presented in Table 2, the NTD ranks are fitted a posteriori on each song individually: for each tolerance value, separately, we select the ranks leading to the best F-measure for the given song. This is called the “oracle ranks” condition, corresponding to the situation where a “perfect” rank selection procedure would exist. Resulting scores provide an (optimistic) performance upper bound.

These two tables exhibit very competitive results. In the first (and most realistic) condition, NTD-based segmentation performance exceeds those of the reference “blind” methods segmentation. In the “oracle ranks” condition, the NTD provides higher F-measures than the state-of-the-art, showing strong potential for the technique, provided an efficient rank selection method is eventually developed.

6. CONCLUSION AND FUTURE WORK

Designing relevant audio features from music remains one of the key questions in many MIR tasks. In this paper, we have proposed a three-way tensor representation of music in frequency, short-term (frames) and mid-term (bars), and means to decompose it under the low-rank Nonnegative Tucker Decomposition (NTD) model. This decomposition turns out to be able to provide a compressed representation of interest, capturing salient patterns in music.

We have illustrated the benefits of the method in a structural segmentation task. The NTD-based representation allows to compute a new type of autosimilarity matrix which exhibits a better contrast than those directly computed on 2D time-frequency representations, and seems well-suited to identify musical patterns at the “right” time-scale for the task. Experimental results are promising and show a potential to compete with state-of-the-art approaches, may they be “blind”, or greedier on training data.

Additional research is required to consolidate the technique. First, as our experiments show, a rank selection criterion would drastically improve segmentation performance. Second, the model does not yet incorporate the notion of proximity between patterns themselves. In parallel, a number of theoretical questions on model identifiability and algorithmic convergence also remain open.

7. REFERENCES

[1] J. B. Smith and M. Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 171–175.

[2] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases," in *ISMIR*, vol. 2, 2002, pp. 287–288.

[3] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2003, pp. 177–180.

[5] A. H. Phan and A. Cichocki, "Extended HALS algorithm for nonnegative tucker decomposition and its applications for multiway analysis and classification," *Neurocomputing*, vol. 74, no. 11, pp. 1956–1969, 2011.

[6] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization," *Neural computation*, vol. 24, no. 4, pp. 1085–1105, 2012.

[7] A. Beck, *First-order methods in optimization*. SIAM, 2017.

[8] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[9] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, "Efficient nonnegative tucker decompositions: Algorithms and uniqueness," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4990–5003, 2015.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[11] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "Tensorly: Tensor learning in python," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 925–930, 2019.

[12] F. Bimbot, G. Sargent, E. Deruty, C. Guichaoua, and E. Vincent, "Semiotic description of music structure: An introduction to the quaero/metiss structural annotations," in *53rd Int. Conf. Audio Engineering Society*, 2014.

[13] G. Sargent, F. Bimbot, and E. Vincent, "Estimating the structural segmentation of popular music pieces under regularity constraints," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 344–358, 2016.

[14] J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[15] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *2000 IEEE Int. Conf. on Multimedia and Expo. ICME2000. Proc. Latest Advances in the Fast Changing World of Multimedia*, vol. 1. IEEE, 2000, pp. 452–455.

[16] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 236–240.

[17] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering," in *ISMIR*, 2014, pp. 405–410.

[18] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *ISMIR*, 2015, pp. 531–537.

[19] MIREX, "Results in the structural segmentation task at the 2015 MIREX contest," 2015. [Online]. Available: https://nema.lis.illinois.edu/nema_out/mirex2015/results/struct/mrx10_2/summary.html

[20] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research," in *ISMIR*, 2016, pp. 547–553.

[21] MIREX, "Results in the structural segmentation task at the 2016 MIREX contest," 2016. [Online]. Available: https://nema.lis.illinois.edu/nema_out/mirex2016/results/struct/mrx10_2/summary.html

[22] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proc. 24th ACM Int. Conf. on Multimedia*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[23] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proc. 14th python in science Conf.*, vol. 8, 2015.

[24] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *ISMIR*, 2014.