ℓ_1-GRAPH BASED MUSIC STRUCTURE ANALYSIS

Yannis Panagakis Constantine Kotropoulos
Dept. of Informatics
Aristotle University of Thessaloniki
Box 451 Thessaloniki, GR-54124, Greece
{panagakis,costas}@aiia.csd.auth.gr

Gonzalo R. Arce
Dept. of Electrical & Computer Engineering
University of Delaware
Newark, DE 19716-3130, U.S.A.
arce@ece.udel.edu

ABSTRACT

An unsupervised approach for automatic music structure analysis is proposed resorting to the following assumption: If the feature vectors extracted from a specific music segment are drawn from a single subspace, then the sequence of feature vectors extracted from a music recording will lie in a union of as many subspaces as the music segments in this recording are. It is well known that each feature vector stemming from a union of independent linear subspaces admits a sparse representation with respect to a dictionary formed by all other feature vectors with nonzero coefficients associated only to feature vectors that stem from its own subspace. Such sparse representation reveals the relationships among the feature vectors and it is used to construct a similarity graph, the so-called ℓ_1-*graph*. Accordingly, the segmentation of audio features is obtained by applying spectral clustering to the ℓ_1-graph. The performance of the just described approach is assessed by conducting experiments on the Pop-Music and the UPF Beatles benchmark datasets. Promising results are reported.

1. INTRODUCTION

A music signal carries a highly structured information at several levels. At the lowest level, a structure is defined by the individual notes, their timbral characteristics, as well as their pitch and time intervals. At an intermediate level, the notes build relatively longer structures, such as melodic phrases, chords, and chord progressions. At the highest level, the structural description of an entire music recording (i.e., its musical form) emerges at the time scale of music sections, such as intro, verse, chorus, bridge, and outro [16, 17].

The musical form of a recording is a high-level information that can be exploited in several music information retrieval (MIR) tasks, including music thumbnailing and summarization [3], chord transcription [12], music semantics learning and music annotation [1], song segment retrieval [1], and remixing [9]. Consequently, the interest in the *automatic music form extraction* or *structure analysis* has increased as is manifested by the considerably amount of research that has been done so far [1, 9, 10, 16, 19]. For a comprehensive review the interested reader is referred to [6, 17] (and the references therein). The majority of methods tested for automatic music structure analysis applies a signal processing stage followed by a representation stage. In the first stage, low-level feature sequences are extracted from the audio signal in order to model its timbral, melodic, and rhythmic content [17]. This is consistent with the findings of Bruderer *et al.*, who state that the perception of structural boundaries in popular music is mainly influenced by the combination of changes in timbre, tonality, and rhythm over the music piece [2]. At the representation stage, a recurrence plot or a similarity matrix is analyzed in order to identify repetitive patterns in the feature sequences by employing hidden Markov models, clustering methods, etc. [6, 17].

In this paper, an unsupervised approach for automatic music structure analysis is proposed. To begin with, each audio recording is represented by a sequence of audio features capturing the variations between the different music segments. Since the music structure is strongly determined by repetition, a similarity matrix should be constructed, that will be analyzed next. Here, the similarity matrix is built by adopting an *one-to-all sparse reconstruction* rather than *one-to-one* (i.e., pairwise) comparisons. To this end, the ℓ_1-graph [5] is constructed in order to capture relationships among the feature vectors. The segmentation of audio features is obtained by applying spectral clustering to the ℓ_1-graph. Apart from the conventional *mel-frequency cepstral coefficients* and *chroma* features, frequently employed in music structure analysis, the *auditory temporal modulations* are also tested here. The performance of the proposed approach is assessed by conducting experiments on two man-

ually annotated benchmark datasets, namely the PopMusic [10] and the UPF Beatles. The experimental results validate the effectiveness of the proposed approach in music structure analysis reaching the performance of the state-of-the-art music structure analysis methods.

The remainder of the paper is as follows. In Section 2, the audio features employed are briefly described. The ℓ_1-graph based music structural analysis framework is detailed in Section 3. Datasets, evaluation metrics, and experimental results are presented in Section 4. Conclusions are drawn and future research directions are indicated in Section 5.

2. AUDIO FEATURE REPRESENTATION

Each 22.050-Hz sampled monaural waveform is parameterized by employing three audio features in order to capture the variations between different music segments. The feature set includes the *auditory temporal modulations* (ATMs), the *mel-frequency cepstral coefficients* (MFCCs), and the *chroma* features.

1) Auditory temporal modulations: ATMs are obtained by modeling the path of human auditory processing. They carry important time-varying information of the music signal [15]. First, by modeling the early auditory system, the acoustic signal is converted into a time-frequency distribution along a logarithmic frequency axis, the so-called *auditory spectrogram*. In this paper, the early auditory system is modeled by employing the Lyons' passive ear model [11]. The derived auditory spectrogram consists of 96 frequency channels ranging from 62 Hz to 11 kHz. The auditory spectrogram is then downsampled along the time axis by a factor of 150 ms, which allows to focus on a more meaningful time-scale for music structural analysis. The underlying temporal modulations of the music signal are derived by applying a wavelet filter along each temporal row of the auditory spectrogram for a set of 8 discrete rates $r \in \{2, 4, 8, 16, 32, 64, 128, 256\}$ Hz ranging from slow to fast temporal rates [15]. Consequently, the entire auditory spectrogram is modeled by a three-dimensional representation of frequency, rate, and time, which is then unfolded along the time-mode in order to obtain a sequence of two-dimensional ATM features.

2) Mel-frequency cepstral coefficients: MFCCs parameterize the rough shape of spectral envelope [13] and thus encode the timbral properties of the music signal, which are closely related to the perception of music structure [2]. Following [16], the MFCCs calculation employs frames of duration 92.9 ms with a hope size of 46.45 ms, and a 42-band filter bank. The correlation between frequency bands is reduced by applying the discrete cosine transform along the log-energies of the bands. The lowest coefficient (i.e., zero-th order) is discarded and the subsequent 12 coefficients form the feature vector that undergoes a zero-mean normalization.

3) Chroma: Chroma features are adept in characterizing the harmonic content of the music signal by projecting the entire spectrum onto 12 bins representing the 12 distinct semitones (or chroma) of a musical octave [13]. They are calculated using 92.9 ms frames with a hope size of 23.22 ms as follows. First, the salience for different fundamental frequencies in the range $80 - 640$ Hz is calculated. The linear frequency scale is transformed into a musical one by selecting the maximum salience value in each frequency range corresponding to one semitone. Finally, the octave equivalence classes are summed over the whole pitch range to yield a 12-dimensional chroma vector.

All the aforementioned features are averaged over the beat (i.e., the basic unit of time in music) frames by employing the beat tracking algorithm described in [8]. Thus a sequence of beat-synchronous feature vectors is obtained.

3. MUSIC STRUCTURE SEGMENTATION BASED ON THE ℓ_1-GRAPH

Since repetition governs the music structure, a common strategy employed is to compare each feature vector of the music recording with all other vectors in order to detect similarities. Let a given audio recording be represented by a feature sequence of N beat frames, i.e., $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. The similarity between the feature vectors is frequently measured by constructing the self-similarity matrix (SDM) $\mathbf{D} \in \mathbb{R}^{N \times N}$ with elements $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$, $i, j \in \{i, 2, \ldots, N\}$, where $d(\cdot, \cdot)$ is a suitable distance metric [9, 16, 17]. Common distance metrics are the Euclidean, $d_E(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ and the cosine distance, $d_C(\mathbf{x}_i, \mathbf{x}_j) = 0.5(1 - \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$, where $\|.\|_2$ denotes the ℓ_2 vector norm. However, the aforementioned approach suffers from two drawbacks: 1) It is very sensitive to noise, since the employed distance metrics are not robust to noise. 2) The resulting SDM is dense and thus it cannot provide the locality information (i.e., to reveal the relationships among neighbor feature vectors that belong to the same segment class), which is valuable in the problem under study.

In order to alleviate the aforementioned drawbacks, we propose to measure the similarities between the feature vectors in an *one-to-all sparse reconstruction* manner rather than to employ the conventional *one-to-one* distance approach by exploiting recent findings in sparse subspace clustering [7].

Formally, let a given audio recording of K music segments be represented by a sequence of N audio feature vectors of size M, i.e., $\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \ldots | \mathbf{x}_N] \in \mathbb{R}^{M \times N}$. By assuming that the feature vectors belonging to the same music segment lie into the same subspace, the columns of $\mathbf{X}$ are drawn from a union of K independent linear subspaces of unknown dimensions. It has been proved that if a feature

vector stems from a union of independent linear subspaces, it admits a sparse representation with respect to a dictionary formed by all other feature vectors with the nonzero coefficients associated to vectors drawn from its own subspace [7]. Therefore, by seeking the sparsest linear combination, the relationship with the other vectors lying in the same subspace is revealed automatically. A similarity graph built from this sparse representation, the so-called ℓ_1-graph [5] is used then in order to segment the columns of $\mathbf{X}$ into K clusters by applying spectral clustering.

Let $\mathbf{X}^i = [\mathbf{x}_1|\mathbf{x}_2|\dots|\mathbf{x}_{i-1}|\mathbf{x}_{i+1}|\dots|\mathbf{x}_N] \in \mathbb{R}^{M\times(N-1)}$. The sparsest solution of $\mathbf{x}_i = \mathbf{X}^i\mathbf{c}$ can be found by solving the optimization problem:

$$\underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}\|_0 \quad \text{subject to } \mathbf{x}_i = \mathbf{X}^i\mathbf{c}, \tag{1}$$

where $\|.\|_0$ is the ℓ_0 quasi-norm returning the number of the non-zero entries of a vector. Finding the solution to the optimization problem (1) is NP-hard due to the nature of the underlying combinational optimization. An approximate solution to the problem (1) can be obtained by replacing the ℓ_0 norm with the ℓ_1 norm as follows:

$$\underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}\|_1 \quad \text{subject to } \mathbf{x}_i = \mathbf{X}^i\mathbf{c}, \tag{2}$$

where $\|.\|_1$ denotes the ℓ_1 norm of a vector. It is well known that if the solution is sparse enough and $M << (N-1)$, then the solution of (1) is equivalent to the solution of (2). The optimization problem (2) can be solved in polynomial time by standard linear programming methods [4]. The well-posedness of (2) relies on the condition $M << (N-1)$, i.e., the sample size must be much larger than the feature dimension. If the ATMs are used to represent audio, the sample size (i.e., the number of beats) is not much larger than the feature vector dimension and thus the just-mentioned condition is violated, because $M = 768$ and $N \approx 500$ on average in the experiments conducted. Accordingly, $\mathbf{c}$ is no longer sparse. To alleviate this problem, it has been proposed to augment $\mathbf{X}^i$ by an $M \times M$ identity matrix and to solve:

$$\underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}\|_1 \quad \text{subject to } \mathbf{x}_i = \mathbf{B}\mathbf{c}, \tag{3}$$

instead of (2), where $\mathbf{B} = [\mathbf{X}^i \mid \mathbf{I}] \in \mathbb{R}^{M\times((N-1)+M)}$ [20].

Since the sparse coefficient vector $\mathbf{c}$ reveals the relationships among $\mathbf{x}_i$ and the feature vectors in $\mathbf{X}^i$, the overall sparse representation of the whole feature sequence $\mathbf{X}$ can be summarized by constructing the weight matrix $\mathbf{W}$ using Algorithm 1. $\mathbf{W}$ can be used to define the so-called ℓ_1-graph [5]. The ℓ_1-graph is a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where the vertices of graph $\mathbf{V}$ are the N audio feature vectors and an edge $(u_i, u_j) \in \mathbf{E}$ exists, whenever $\mathbf{x}_j$ participates in the sparse representation of $\mathbf{x}_i$. Accordingly, the adjacency matrix of $\mathbf{G}$ is $\mathbf{W}$. Unlike the conventional SDM, the adjacency matrix $\mathbf{W}$ is robust to noise. The ℓ_1-graph $\mathbf{G}$ is an unbalanced digraph. A balanced graph $\hat{\mathbf{G}}$ can be built with adjacency matrix $\hat{\mathbf{W}}$ with elements $\hat{w}_{ij} = 0.5\,(|w_{ij}| + |w_{ji}|)$, where $|.|$ denotes the absolute value. $\hat{\mathbf{W}}$ is still a valid representation of the similarity between the features vectors, since if $\mathbf{x}_i$ can be expressed as a compact linear combination of some feature vectors including $\mathbf{x}_j$ (all from the same subspace or music segment here), then $\mathbf{x}_j$ can also be expressed as a compact linear combination of feature vectors in the same subspace including $\mathbf{x}_i$ [7]. In Figure 1, the $\hat{\mathbf{W}}$ is depicted for the three features tested. It can be seen that $\hat{\mathbf{W}}$ has a block structure for the ATMs, while it is unstructured and more dense for the MFCCs and the Chroma features. This observation validates that the main assumptions made in the paper hold here for the ATMs, but not for the MFCCs and the Chroma features.

Algorithm 1 ℓ_1-Graph Construction [5].

Input: Audio feature sequence $\mathbf{X} \in \mathbb{R}^{M\times N}$.
Output: Weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$.

```
for i = 1 → N do
    B = [X^i | I].
    argmin_c ||c||_1   subject to x_i = Bc.
    for j = 1 → N do
        if j < i then
            w_ij = c_j.
        else
            w_ij = c_{j-1}.
        end if
    end for
end for
```

The segmentation of the audio feature vectors can be obtained by spectral clustering algorithms, such as the normalized cuts [18] as illustrated in Algorithm 2.

Algorithm 2 Music Segmentation via ℓ_1-Graph.

Inputs: Audio feature sequence $\mathbf{X} \in \mathbb{R}^{M\times N}$ and number of segments K.
Output: Audio feature sequence segmentation.

1: Obtain the adjacency matrix $\mathbf{W}$ of ℓ_1-graph by Algorithm 1.
2: Build the symmetric adjacency matrix of the ℓ_1-graph $\hat{\mathbf{G}}$: $\hat{\mathbf{W}} = 0.5 \cdot (|\mathbf{W}| + |\mathbf{W}^T|)$.
3: Employ normalized cuts [18] to segment the vertices of $\hat{\mathbf{G}}$ into K clusters.

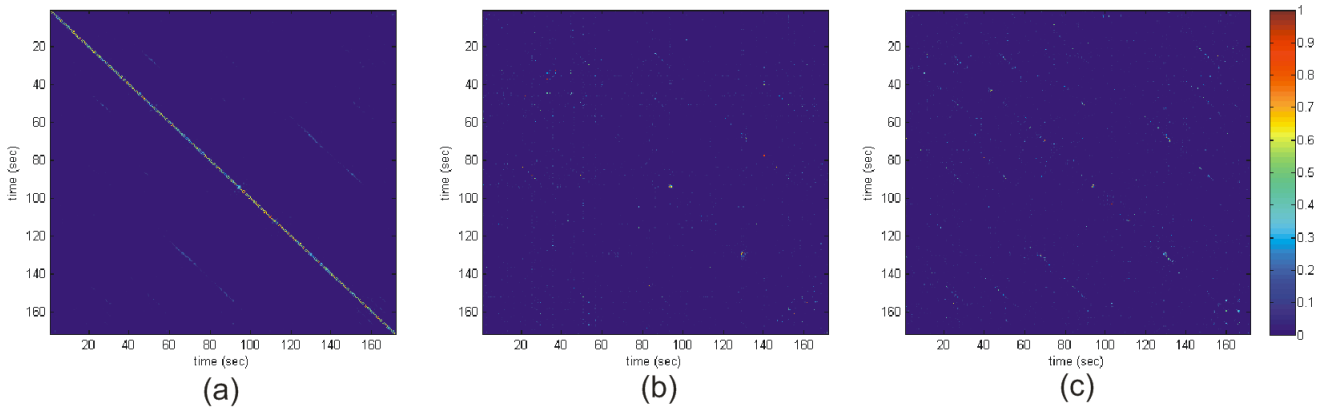

Figure 1. The adjacency matrix $\hat{\mathbf{W}}$ of the ℓ_1-graph for the song "*I saw her standing there*" by The Beatles for (a) the ATMs, (b) the MFCCs, and (c) the Chroma features.

4. EXPERIMENTAL EVALUATION

The performance of the proposed music structure analysis approach is assessed by conducting experiments on two manually annotated datasets of Western popular music pieces. Several evaluation metrics are employed to assess system performance from different points of view.

4.1 Datasets

PopMusic dataset [10]: The dataset consists of 60 music recordings of rock, pop, hip-hop, and jazz. Half of the recordings originate from a variety of well-known artists appeared the past 40 years, including Britney Spears, Eminem, Madonna, Nirvana, etc. This subset is abbreviated as *Recent* hereafter. The remaining 30 music recordings are by The Beatles. The ground-truth segmentation of each song contains between 2 and 15 different segments classes. The number of classes is 6, while each recording is found to contain 11 segments on average [1, 10]. The subset contains the Beatles recordings is referred ta as *Beatles*.

UPF Beatles dataset: [1] The dataset consists of 174 songs by The Beatles that are annotated by the musicologist Alan W. Pollack. Segmentation time stamps were inserted at Universitat Pompeu Fabra (UPF) as well. Each music recording contains on average 10 segments from 5 unique classes [19]. Since all the recordings are from the same band, there is less variation in the music style and the timbral characteristics than the other datasets.

4.2 Evaluation Metrics

Following [1, 9, 10, 16, 19], the segment labels are evaluated by employing the pairwise F-measure, which is one of the standard metrics of clustering quality. It compares pairs of beats, which are assigned to the same cluster by music structure analysis against the reference segmentation. Let $\mathbb{F}_A$ be the set of similarly labeled pairs of beats in a recording according to the music structure analysis algorithm and $\mathbb{F}_H$ be the set of similarly labeled pairs in the human reference segmentation. The pairwise precision, $P_{pairwise}$, the pairwise recall, $R_{pairwise}$, and the pairwise F-measure, $F_{pairwise}$, are defined as follows: $P_{pairwise} = \frac{|\mathbb{F}_A \cap \mathbb{F}_H|}{|\mathbb{F}_A|}$, $R_{pairwise} = \frac{|\mathbb{F}_A \cap \mathbb{F}_H|}{|\mathbb{F}_H|}$, and $F_{pairwise} = 2 \cdot \frac{P_{pairwise} \cdot R_{pairwise}}{P_{pairwise} + R_{pairwise}}$, where $|.|$ denotes the set cardinality. The average number of segments per song in each dataset is reported as well.

The segment boundary detection is evaluated separately by employing the standard precision, recall, and F-measure. Following [1, 10, 16], a boundary detected by the proposed approach is considered correct, if it falls within some fixed small distance δ away from the reference boundary. Each reference boundary can be retrieved by at most one output boundary. Let $\mathbb{B}_A$ and $\mathbb{B}_H$ denote the sets of segment boundaries according to the music structure analysis algorithm and the human reference, respectively. Then, $P = \frac{|\mathbb{B}_A \cap \mathbb{B}_H|}{|\mathbb{B}_A|}$, $R = \frac{|\mathbb{B}_A \cap \mathbb{B}_H|}{|\mathbb{B}_H|}$, and $F = 2 \cdot \frac{P \cdot R}{P + R}$. The parameter δ is set to 3 s in our experiments as was also done in [1, 10, 16].

4.3 Experimental Results

The structural segmentation is obtained by applying the proposed approach to various feature sequences. Following the experimental setup employed in [1, 9, 10, 16, 19], the number of clusters K was set to 6 for the PopMusic dataset, while $K = 4$ for the UPF Beatles dataset. For comparison purposes, experiments are conducted by applying the normalized cuts [18] apart from the ℓ_1-graph and the SDM with the Euclidean distance computed for the three audio features. The segment-type labeling performance for the PopMusic and the UPF Beatles datasets is summarized in Table 1 and

[1] http://www.dtic.upf.edu/ perfe/annotations/sections/license.html

Table 2, respectively.

Method/Reference	Dataset	$F_{pairwise}$	Av. Number of Segments
ATM + ℓ_1-graph based segmentation	Beatles	**0.6140**	8.8333
	Recent	**0.5885**	12.6087
	PopMusic	**0.5912**	11.8679
MFCCs + ℓ_1-graph based segmentation	Beatles	0.4029	199.3667
	Recent	0.3884	248.2826
	PopMusic	0.3966	239.6316
Chroma + ℓ_1-graph based segmentation	Beatles	0.4191	153.7667
	Recent	0.3520	260.3043
	PopMusic	0.3900	200
ATM + SDM based segmentation	Beatles	0.4243	145.7000
	Recent	0.3975	141.3913
	PopMusic	0.4027	125.5283
MFCCs + SMD based segmentation	Beatles	0.3664	226.3667
	Recent	0.3663	305.9130
	PopMusic	0.3664	260.8868
Chroma + SDM based segmentation	Beatles	0.3499	220.4333
	Recent	0.3312	276.1739
	PopMusic	0.3418	244.6226
MFCCs unconstrained [1]	PopMusic	0.577	17.9
MFCCs constrained [1]	PopMusic	**0.620**	10.7
Chroma constrained [1]	PopMusic	0.51	12
K-means clustering [10]	Beatles	0.425	N/A
	Recent	0.457	N/A
	PopMusic	0.441	N/A
Mean-field clustering [10]	Beatles	0.538	N/A
	Recent	0.560	N/A
	PopMusic	0.549	N/A
Constrained clustering [10]	Beatles	0.604	N/A
	Recent	0.605	N/A
	PopMusic	**0.603**	N/A

Table 1. Segment-type labeling performance on the PopMusic dataset.

By inspecting Tables 1 and 2, it is clear that the ℓ_1-graph based segmentation outperforms the SDM based segmentation in terms of pairwise F-measure for all the audio features employed in both datasets. Moreover, the ATMs offer a parsimonious representation for the task of music structure analysis, especially when employed in the construction of the ℓ_1-graph.

The best results reported for segment-type labeling on the PopMusic dataset are obtained here, when the ATMs are employed for audio representation and the segmentation is performed on the ℓ_1-graph defined by them. These results are comparable to the best reported results by Levy and Sandler [10], while inferior to those reported by Barrington *et al.* [1]. It is worth noting that the clustering is performed without any constraints in the proposed approach, which is not the case for the best results reported in [1, 10]. In an unconstrained clustering setting, the proposed system outperforms the systems discussed in [1, 10].

Method/Reference	$F_{pairwise}$	Av. Number of Segments
ATM + ℓ_1-graph based segmentation	**0.5938**	8.5215
MFCCs + ℓ_1-graph based segmentation	0.4664	181.9950
Chroma + ℓ_1-graph based segmentation	0.4563	116.2989
ATM + SDM based segmentation	0.4711	81.0376
MFCCs + SDM based segmentation	0.3985	190.5489
Chroma + SDM based segmentation	0.4066	167.9239
Method in [10] as evaluated in [16]	0.584	N/A
[16]	0.599	N/A
[19]	**0.600**	N/A
[9]	**0.621**	N/A

Table 2. Segment-type labeling performance on the UPF Beatles dataset.

Method/Reference	Dataset	F	P	R
ATM + ℓ_1-graph based segmentation	PopMusic	0.5227	0.4737	0.6274
MFCCs constrained [1]	PopMusic	**0.610**	0.620	0.650
Chroma constrained [1]	PopMusic	0.420	0.410	0.460
EchoNest reported in [1]	PopMusic	0.450	0.410	0.560
K-means clustering [10]	PopMuic	0.437	0.809	0.311
Mean-field clustering [10]	PopMusic	0.448	0.366	0.665
Constrained clustering [10]	PopMusic	0.590	0.648	0.567
ATM + ℓ_1-graph based segmentation	UPF Beatles	0.5304	0.5338	0.5670
Method in [10] as evaluated in [16]	UPF Beatles	**0.612**	0.600	0.646
[16]	UPF Beatles	0.55	0.521	0.612
Timbre [9]	UPF Beatles	0.586	0.581	0.619
Chroma [9]	UPF Beatles	0.500	0.465	0.522
Timbre & Chroma [9]	UPF Beatles	0.536	0.49	0.55

Table 3. Boundary detection performance on the PopMusic and the UPF Beatles dataset.

In the UPF Beatles dataset, the best results for segment-type labeling are obtained again when the ATMs are employed for audio representation and the segmentation is perform on the ℓ_1-graph constructed using $\hat{\mathbf{W}}$. The reported results are comparable to those obtained by the state-of-the-art music structure analysis on this dataset [16, 19]. The proposed approach is not directly comparable to that in [9] due to the use of slightly different reference segmentations.

The average number of segments detected by our approach is 11.86 and 8.52, when according to the ground-truth the actual average number of segments is 11 and 10 for the PopMusic and the UPF Beatles dataset, respectively. This result is worth noting since no constraints have been enforced during clustering.

The performance of the proposed approach deteriorates when either the MFCCs or the chroma features are employed for music representation. The low pairwise F-measure and the over-segmentation can be be attributed to the fact that the underlying assumptions set in Section 3 do not hold for such representations.

Since the performance of our approach is clearly inferior when MFCCs or chroma features are used for music representation, only the ATMs are employed in the segment-boundary detection task. The boundary detection results are summarized in Table 3 for both the PopMusic and the UPF Beatles datasets. EchoNest refers to the commercial online music boundary detection service provided by The Echonest and evaluated in [1]. By inspecting Table 3 the proposed approach is clearly inferior to the system proposed by Levy and Sandler [10] for music boundary detection on both datasets. The success of the latter approach can be attributed to the constraints imposed during clustering. Consequently, the results obtained by the proposed approach in music boundary detection could be considered as acceptable, since the performance of our system is rated above that reported for many other state-of-the-art systems with or without constraints (e.g., the EchoNest online service). It is worth mentioning that neither of the methods appearing in Table 3 reaches the accuracy of the specialized bound-

ary detection methods (e.g., that in [14]) which achieves a boundary F-measure of 0.75 on a test set similar to the Beatles subset of the PopMusic dataset. However, such boundary detection methods, do not model the music structure and provide no characterization of the segments between the boundaries as the proposed approach as well as the methods in [1, 9, 10, 16, 19] do.

5. CONCLUSIONS

A novel unsupervised music structure analysis approach has been proposed. This framework resorts to ATMs for music representation, while the segmentation is performed by applying spectral clustering on the ℓ_1-graph. The performance of the proposed approach is assessed by conducting experiments on two benchmark datasets. The experimental results on music structure analysis are comparable to those reported by other state-of-the-art music structure analysis systems. Moreover, promising results on music boundary detection are reported. It is believed that by imposing constraints during clustering in the proposed approach both the music structure analysis and the music boundary detection will be considerably improved. This point will be investigated in the future. Another future research direction is to automatically detect the number of music segments.

Acknowledgements

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heraclitus II. Investing in Knowledge Society through the European Social Fund.

6. REFERENCES

[1] L. Barrington, A. Chan, and G. Lanckriet. Modeling music as a dynamic texture. *IEEE Trans. Audio, Speech, and Language Processing*, 18(3):602–612, 2010.

[2] M. Bruderer, M. McKinney, and A. Kohlrausch. Structural boundary perception in popular music. In *Proc. 7th Int. Symposium Music Information Retrieval*, pages 198–201, Victoria, Canada, 2006.

[3] W. Chai and B. Vercoe. Structural analysis of musical signals for indexing and thumbnailing. In *Proc. ACM/IEEE Joint Conf. Digital Libraries*, pages 27–34, 2003.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[5] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with l1-graph for image analysis. *IEEE Trans. Image Processing*, 19(4):858–866, 2010.

[6] R. B. Dannenberg and M. Goto. Music structure analysis from acoustic signals. In D. Havelock, S. Kuwano, and M. Vorländer, editors, *Handbook of Signal Processing in Acoustics*, pages 305–331. Springer, New York, N.Y., USA, 2008.

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 2790–2797, Miami, FL, USA, 2009.

[8] D. Ellis. Beat tracking by dynamic programming. *J. New Music Research*, 36(1):51–60, 2007.

[9] F. Kaiser and T. Sikora. Music structure discovery in popular music using non-negative matrix factorization. In *Proc. 11th Int. Symposium Music Information Retrieval*, pages 429–434, Utrecht, Netherlands, 2010.

[10] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Trans. Audio, Speech, and Language Processing*, 16(2):318–326, 2008.

[11] R. Lyon. A computational model of filtering, detection, and compression in the cochlea. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 1282–1285, Paris, France, 1982.

[12] M. Mauch, K. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. 10th Int. Symposium Music Information Retrieval*, pages 231–236, Kobe, Japan, 2009.

[13] M. Müller, D. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE J. Sel. Topics in Signal Processing (accepted for publication)*, 2011.

[14] B. Ong and P. Herrera. Semantic segmentation of music audio contents. In *Proc. Int. Computer Music Conference*, Barcelona, Spain, 2005.

[15] Y. Panagakis, C. Kotropoulos, and G. R. Arce. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Trans. Audio, Speech, and Language Technology*, 18(3):576–588, 2010.

[16] J. Paulus and A. Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Trans. Audio, Speech, and Language Processing*, 17(6):1159–1170, 2009.

[17] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In *Proc. 11th Int. Symposium Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[19] R. Weiss and J. Bello. Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. In *Proc. 11th Int. Symposium Music Information Retrieval*, pages 123–128, Utrecht, Netherlands, 2010.

[20] J. Wright and Y. Ma. Dense error correction via l1-minimization. *IEEE Trans. Information Theory*, 56(7):3540–3560, 2010.