MUSIC STRUCTURE ANALYSIS BY SUBSPACE MODELING

Yannis Panagakis and Constantine Kotropoulos

Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, GREECE
email: {panagakis,costas}@aiia.csd.auth.gr

ABSTRACT

Automatic music structure analysis is casted as a subspace clustering problem. By assuming that the feature vectors extracted from a specific music segment are drawn from a single subspace, any sequence of such feature vectors derived from a music recording will lie in a union of as many subspaces as the music segments in the recording are. First, the sparse and the low-rank subspace clustering is tested for music structure analysis by employing three types of beat-synchronous audio feature sequences. Next, a novel computational efficient subspace clustering method is proposed, that is coined as *ridge representation subspace clustering* (RRSC). The performance of the aforementioned three subspace clustering methods is assessed by conducting experiments on the manually annotated Beatles benchmark dataset. The experimental results indicate that: 1) the performance of the RRSC is comparable or exceeds that of the sparse and the low-rank subspace clustering and 2) the RRSC outperforms the state-of-the-art methods proposed for music structure analysis.

Index Terms— Music Structure Analysis, Music Segmentation, Subspace Clustering, Ridge Regression.

1. INTRODUCTION

The *musical form* of a music piece refers to the structural description of the piece at the time scale of segments, such as intro, verse, chorus, bridge, etc. [1]. Its deduction from the audio signal is known as *automatic music structure analysis*. The latter is a core task in music thumbnailing and summarization, chord transcription , learning of music semantics and music annotation [2], song segment retrieval [2], and remixing [3].

Human listeners are able to analyze and segment the music into meaningful parts by detecting the structural boundaries between the segments based on perceived changes in timbre, tonality, and rhythm over the music piece [4]. Automatic music structure analysis employs low-level feature sequences, extracted from the audio signal, in order to model the timbral, melodic, and rhythmic content [1]. The segmentation of the feature sequences into structural parts is performed by employing methods based on either repetition, homogeneity, or novelty to analyze a recurrence plot or a self-similarity distance matrix [1–3,5–8]. For a comprehensive review on automatic music structure analysis systems, the interested reader is referred to [1] (and the references therein).

In this paper, motivated by our previous work [6], music structure analysis is casted as a subspace clustering problem [9–11]. Three types of audio features, namely the *mel-frequency cepstral coefficients* (MFCCs), the *chroma* features, and the *auditory temporal modulations* (ATMs) are employed in order to form beat-synchronous feature sequences modeling the audio signal. Due to the timbral, tonal, and rhythmic homogeneity within the music segments, it is reasonable to assume that the audio features extracted from a specific music segment are highly correlated and thus linearly dependent. Therefore, there is a linear subspace that spans the beat-synchronous audio features for any music segment implying that the sequence of feature vectors extracted from the whole music recording will lie in a union of as many independent linear subspaces as the music segments of this recording are. Consequently, under this assumption, the segmentation of music can be performed by applying any subspace clustering method [9] on the features sequences. State-of-the-art subspace clustering methods are seeking the *sparsest representation* (SR) [10] or the *lowest-rank representation* (LRR) [11] of all the beat-synchronous audio features collectively. Such representations exhibit nonzero within-subspace affinities and almost zero between-subspace affinities. Hence, a suitable sparse or low-rank affinity matrix for segmentation can be constructed. Despite their effectiveness in practical applications [9–11], both the SR-based (i.e., the *sparse subspace clustering* (SSC) [10]) and the LRR-based subspace clustering [11] are computationally demanding methods. Accordingly, they are not suitable for large-scale problems, such as the music segmentation involving hundreds of high dimensional beat-synchronous audio features per music recording.

To remedy this drawback, a novel computationally efficient subspace clustering method is proposed that employs the *ridge representation* (RR) of all the audio features collectively. To this end, a *ridge regression* problem is solved. The

RR shares the same advantage with the SR and the LRR. That is, when the data are noiseless (i.e., come exactly from a union of independent linear subspaces), it can be proved that the RR has nonzero within-subspace affinities and zero between-subspace affinities. Moreover, unlike the SR and the LRR, the RR is unique, and admits a closed form. Having derived the RR-based affinity matrix, the application of spectral clustering to this affinity matrix reveals finally the segmentation of the feature sequence into music segments (or subspaces in general). The proposed method is referred to as *ridge representation subspace clustering* (RRSC).

The performance of subspace clustering methods in music structure analysis is assessed by conducting experiments on the manually annotated Beatles benchmark dataset. The experimental results demonstrate the effectiveness of the proposed RRSC over the SSC and the LRR. Furthermore, the RRSC outperforms the state-of-the-art music structure analysis methods.

2. AUDIO FEATURE REPRESENTATION

The variations between different music segments are captured by extracting three audio features from each 22.05-kHz sampled monaural music recording. In particular, the MFCCs, the Chroma features and the ATMs are employed.

1) The MFCCs encode the timbral properties of the music signal by parameterizing the rough shape of spectral envelope. Following [7], the MFCC extraction employs frames of duration 92.9 ms with a hop size of 46.45 ms, and a 42-band filter bank. The correlation between the frequency bands is reduced by applying the discrete cosine transform along the log-energies of the bands. The zeroth order coefficient is discarded yielding a sequence of 12-dimensional MFCC vectors.

2) The Chroma features are able to characterize 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. They are calculated by employing 92.9 ms frames with a hop size of 23.22 ms as follows. First, the salience of 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 sequence of 12-dimensional chroma vectors.

3) The auditory temporal modulations carry important time-varying information of the audio signal [12]. They are obtained by modeling the path of human auditory processing as a two-stage process. In the first stage, which models the early auditory system, the acoustic signal is converted into a time-frequency distribution along a logarithmic frequency axis, the so-called *auditory spectrogram*. The early auditory system is modeled by Lyons' passive ear model [13] employing 96 frequency channels ranging from 62 Hz to 11 kHz. The auditory spectrogram is then downsampled along the time axis in order to obtain 10 feature vectors between two successive beats. The underlying temporal modulations of the music signal are derived by applying a biorthogonal wavelet filter along each temporal row of the auditory spectrogram, where its mean has been previously subtracted, for 8 discrete rates $r \in \{2, 4, 8, 16, 32, 64, 128, 256\}$ Hz ranging from slow to fast temporal rates. Thus, the entire auditory spectrogram is modeled by a three-dimensional representation of frequency, rate, and time which is then unfolded[1] along the time-mode in order to obtain a sequence of $96 \times 8 = 728$-dimensional ATMs features.

Postprocessing. Sequences of *beat-synchronous* feature vectors are obtained by averaging any feature sequence over the beat frames using the beat tracking algorithm described in [14]. Each row of the beat-synchronous feature matrix is filtered by applying an average filter of length 8. Finally, each feature vector undergoes a normalization in order to have zero-mean and unit ℓ_2 norm.

3. MUSIC STRUCTURE ANALYSIS BASED ON SUBSPACE MODELING

Let a given music recording of K music segments be represented by a sequence of N beat-synchronous audio feature vectors of size d, i.e., $\mathbf{X} = [\mathbf{x}_1|\mathbf{x}_2|\dots|\mathbf{x}_N] \in \mathbb{R}^{d \times N}$. Assume that the feature vectors belong to a certain music segment lie into the same subspace. Then, the columns of $\mathbf{X}$ are drawn from a union of K independent linear subspaces of unknown dimensions. Three methods are discussed for the derivation of the affinity matrix of the subspaces that are based on sparse, low-rank, and ridge representation. Next, music structure analysis can be obtained by applying spectral clustering on these affinity matrices.

3.1. Subspace Clustering by Sparse Representation

Elhamifar and Vidal have proved that if a feature vector stems from a union of independent linear subspaces, it admits a sparse representation with respect to the dictionary formed by all other feature vectors. In particular, the nonzero coefficients are associated to vectors drawn from its own subspace [10]. Therefore, by seeking the sparsest linear combination, the relationship with the other vectors lying in the same subspace is revealed automatically. Indeed, the sparse representation matrix $\mathbf{Z} \in \mathbb{R}^{N \times N}$ containing the sparse coefficients in its columns can be found by solving the convex problem:

$$\underset{\mathbf{Z}}{\text{argmin}} \quad \|\mathbf{Z}\|_1 \quad \text{s.t. } \mathbf{X} = \mathbf{X}\,\mathbf{Z}, \;\; \text{diag}(\mathbf{Z}) = \mathbf{0}, \qquad (1)$$

[1] The tensor unfolding can be implemented in Matlab by employing the `tenmat` function of the MATLAB Tensor Toolbox available at: `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`.

where $\|\mathbf{Z}\|_1 = \sum_i \sum_j |z_{ij}|$ is the matrix ℓ_1-norm, $|.|$ is the absolute value operator, and diag($\mathbf{Z}$) returns a vector which contains the diagonal elements of $\mathbf{Z}$.

A sparse nonnegative symmetric affinity matrix $\mathbf{W} \in \mathbb{R}_+^{N \times N}$ can be constructed having elements $w_{ij} = 0.5(|z_{ij}| + |z_{ji}|)$ [10]. The columns of $\mathbf{X}$ can then be segmented into K clusters by applying the normalized cuts [15] onto the sparse affinity matrix $\mathbf{W}$. This method is referred to as *sparse subspace clustering* (SSC) [10].

3.2. Subspace Clustering by Low-Rank Representation

Another suitable representation for subspace clustering is the the low-rank representation of all the features jointly. Ideally, the LRR exhibits nonzero within-subspace affinities and zero between-subspace affinities. The LRR matrix can be obtained by solving the following convex optimization problem [11]:

$$\underset{\mathbf{Z},\mathbf{E}}{\operatorname{argmin}} \quad \|\mathbf{Z}\|_* \quad \text{s.t. } \mathbf{X} = \mathbf{X}\,\mathbf{Z}. \tag{2}$$

$\|\mathbf{Z}\|_*$ denotes the nuclear norm of $\mathbf{Z}$. That is, the sum of its singular values. When the data contain outliers , the LRR matrix can be found by solving:

$$\underset{\mathbf{Z},\mathbf{E}}{\operatorname{argmin}} \quad \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{2,1} \quad \text{s.t. } \mathbf{X} = \mathbf{X}\,\mathbf{Z} + \mathbf{E}. \tag{3}$$

$\|\mathbf{E}\|_{2,1} = \sum_j \sqrt{\sum_i e_{ij}^2}$ is the ℓ_2/ℓ_1-norm of $\mathbf{E}$.

Let $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition (SVD) of $\mathbf{Z}$, $\tilde{\mathbf{U}} = \mathbf{U}(\mathbf{\Sigma})^{\frac{1}{2}}$, and $\mathbf{M} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T$. Then, a low-rank nonnegative symmetric affinity matrix $\mathbf{W} \in \mathbb{R}_+^{N \times N}$ can be constructed with elements [11]:

$$w_{ij} = m_{ij}^2. \tag{4}$$

The segmentation of the data (i.e., the columns of $\mathbf{X}$) into K clusters is performed by employing the normalized cuts [15] onto the low-rank affinity matrix $\mathbf{W}$. This method is known as LRR-based subspace clustering [11].

3.3. Subspace Clustering by Ridge Representation

Despite the effectiveness of the SSC and the LRR-based subspace clustering, both methods need to solve computationally demanding optimization problems via iterative algorithms in order to derive the representation matrix $\mathbf{Z}$. In particular, the SSC involves a constrained LASSO regression. Such problems are non-smooth and thus a high computational effort is needed for their solution. For the LRR, although the augmented Lagrange multiplier method is a powerful means to solve (3), one SVD is required at each iteration and in practice the algorithm converges after hundreds of iterations. Consequently, both the SSC and the LRR are not suitable for large-scale problems.

In practice, one would like to learn *efficiently* the representation matrix $\mathbf{Z} \in \mathbb{R}^{N \times N}$, such that $\mathbf{X} = \mathbf{X}\,\mathbf{Z}$, with $z_{ij} = 0$, if $\mathbf{x}_i$ and $\mathbf{x}_j$ lie in different subspaces and nonzero otherwise. Such a representation matrix $\mathbf{Z}$ can be found by solving a least-squares problem with Frobenius norm regularization, the so-called *ridge regression* problem:

$$\underset{\mathbf{Z}}{\operatorname{argmin}} \quad \|\mathbf{X} - \mathbf{X}\,\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}\|_F^2. \tag{5}$$

In (5) $\|.\|_F$ denotes the Frobenius norm. The unique solution of the unconstrained convex problem (5) is referred to as *ridge representation* (RR) matrix and it is given in closed-form by:

$$\mathbf{Z} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{X}). \tag{6}$$

Technically, the desired property of the RR matrix to admit nonzero entries for within-subspace affinities and zero entries for between-subspace affinities is enforced by the regularization term $\lambda\|\mathbf{Z}\|_F^2$ in (5) as proved in Theorem 1, which is a consequence of Lemma 1. The proof of Theorem 1 is omitted due to lack of space.

Lemma 1 [16]. For any four matrices $\mathbf{B}, \mathbf{C}, \mathbf{D}$, and $\mathbf{F}$ of compatible dimensions,

$$\left\| \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{F} \end{bmatrix} \right\|_F^2 \geq \left\| \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \right\|_F^2 = \|\mathbf{B}\|_F^2 + \|\mathbf{F}\|_F^2. \tag{7}$$

Theorem 1. Assume the columns of $\mathbf{X}$ (i.e., feature vectors) are drawn from a union of K linear independent subspaces of unknown dimensions and without loss of generality, $\mathbf{X} = [\mathbf{X}_1|\mathbf{X}_2|\ldots|\mathbf{X}_K] \in \mathbb{R}^{d \times N}$, where the columns of $\mathbf{X}_k \in \mathbb{R}^{d \times N_k}$, $k = 1, 2, \ldots, K$ correspond to the N_k feature vectors originating from the kth subspace. The minimizer of (5) is block-diagonal.

As in LRR, a nonnegative symmetric affinity matrix $\mathbf{W} \in \mathbb{R}_+^{N \times N}$ can by constructed by computing the SVD of $\mathbf{Z}$ with elements as in (4). Again, the columns of $\mathbf{X}$ are partitioned into K clusters (i.e., music segments here) by applying the normalized cuts [15] onto the RR-based affinity matrix. The aforementioned fast subspace clustering method is referred to as *ridge regression subspace clustering* (RRSC).

4. EXPERIMENTAL EVALUATION

4.1. Dataset, Evaluation Procedure, and Evaluation Metrics

Beatles dataset[2]: The dataset consists of 180 songs by The Beatles. The songs are annotated by the musicologist Alan W. Pollack. Segmentation time stamps were inserted at the Universitat Pompeu Fabra (UPF). Each music recording contains on average 10 segments from 5 unique segment classes (i.e., intro, verse, chorus etc.) [8].

The structure segmentation is obtained by applying the SSC, the LRR, and the RRSC to the three feature sequences.

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

As a reference method for structure segmentation, the normalized cuts [15] are applied to the *self-distance matrix* (SDM) constructed by employing the cosine distance for the three audio features described in Section 2. Two sets of experiments were conducted on the Beatles dataset. First, following the experimental setup employed in [2, 3, 5–8, 17], the number of clusters (i.e., segments) K was considered constant and equal to 4. In the second experiment, the number of segments was estimated by employing the soft-thresholding approach [11] for each music recording. That is, the number of segments $\bar{K}$ is estimated by:

$$\bar{K} = N - \text{int}(\sum_{i=1}^{N} f_\tau(\sigma_i)). \quad (8)$$

The function int(.) returns the nearest integer of a real number, $\{\sigma_i\}_{i=1}^{N}$ denotes the set of the singular values of the Laplacian matrix derived by the corresponding affinity matrix, and f_τ is the soft-thresholding operator defined as $f_\tau(\sigma) = 1$ if $\sigma \geq \tau$ and $\log_2(1 + \frac{\sigma^2}{\tau^2})$, otherwise. The threshold $\tau \in (0, 1)$. The optimal values of the various parameters (i.e., λ, τ) were determined by a grid search over 10 randomly selected music recordings of the dataset.

Following [2, 3, 5–8, 17], 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 identically labeled pairs of beats in a recording according to the music structure analysis algorithm and $\mathbb{F}_H$ be the set of identically labeled pairs in the human reference segmentation. The pairwise precision, PP, the pairwise recall, PR, and the pairwise F-measure, PF, are defined as: $PP = \frac{|\mathbb{F}_A \cap \mathbb{F}_H|}{|\mathbb{F}_A|}$, $PR = \frac{|\mathbb{F}_A \cap \mathbb{F}_H|}{|\mathbb{F}_H|}$, and $PF = 2 \cdot \frac{PP \cdot PR}{PP + PR}$, where $|.|$ denotes the set cardinality.

4.2. Experimental Results

The segment-type labeling performance for the Beatles dataset is summarized in Table 1. In order to improve the segment-type labeling performance, the SDM-, SR-, LRR-, and RR-based affinity matrices were post-processed as follows. Let any affinity matrix $\mathbf{W}$ be decomposed as $\mathbf{W} = \mathbf{D}_w + \mathbf{U}_w + \mathbf{L}_w$, where $\mathbf{D}_w$ contains the 5 main diagonals of $\mathbf{W}$ while $\mathbf{U}_w$ and $\mathbf{L}_w$ are the upper and the lower triangular matrices of $\mathbf{W} - \mathbf{D}_w$, respectively. Next, the aforementioned three matrices are normalized by dividing their elements with the maximum element. Let us denote by $\hat{\mathbf{D}}_w$, $\hat{\mathbf{U}}_w$, and $\hat{\mathbf{L}}_w$ the resulting normalized matrices in the range $[0, 1]$ of $\mathbf{D}_w$, $\mathbf{U}_w$, and $\mathbf{L}_w$, respectively. Consequently the normalized affinity $\hat{\mathbf{W}}$ is obtained, which is then filtered with a 2D Gabor filter with angle $\pi/4$. The results obtained by applying the normalized cuts to the post-processed affinity matrix are shown in the columns of tables indicated as (w/P).

Let us begin with a fixed number of segments (i.e., $K =$) By inspecting Table 1, one can see that the three subspace clustering methods, namely the SSC, the LRR, and the proposed RRSC outperform the conventional SDM based music structure analysis in terms pairwise F-measure for the all the features. The RRSC and the SSC outperform the LRR, while in the most cases the RRSC outperforms the subspace clustering that is compare to. Another advantage of the RRSC compared to the SSC and the LRR is its computational efficiency. The average CPU time for the calculation of the RR-based affinity matrix is 0.858 CPU seconds, while the SSC and the LRR need 42.160 and 193.445 CPU seconds, respectively. The postprocessing of the affinity matrix not only improves the clustering performance, but reduces also the number of segments. Interestingly, the number of segments is close to 10 (i.e., the actual average number of segments according to the ground-truth), when the ATMs are employed for audio representation. This result is worth noting, since no constraints have been enforced during clustering. The best results reported for segment-type labeling on the Beatles dataset are obtained here, when the ATMs are employed for audio representation and the segmentation is performed by the RRSC. These results outperform those obtained by the state-of-the-art music segmentation methods listed in the last five rows of Table 1.

The segment-type labeling performance for the Beatles dataset by employing the automatic estimation of K using (8) is reported in the last five columns of Table 1. Again, the clustering performance of the RRSC is comparable or even better than that of the SSC and the LRR. When the ATMs are employed for audio representation and the segmentation is performed by the RRSC on the postprocessed affinity matrix, the pairwise F-measure is 0.60. Chen *et al.* have reported a pairwise F-measure equal to 0.63 by automatically estimating the number of segments (i.e., K) [17]. These results indicate that it is possible to perform a robust unsupervised music structure analysis in a fully automatic setting.

5. CONCLUSIONS

In this paper, it has been demonstrated that music structure analysis can be modeled as a subspace clustering problem. Thus, it can be solved effectively by subspace clustering methods. To this end, the SSC and the LRR have been applied to three beat-synchronous audio features for music structure analysis. Moreover, a novel subspace clustering method (i.e., the RRSC) that builds on the ridge regression of all the features jointly has been proposed. The experimental results on the Beatles dataset indicate the power of the RRSC for music structure analysis. In particular, when the ATMs are employed for audio representation, the RRSC yields a state-of-the-art performance in music structure analysis.

		Fixed $K = 4$					Automatically estimated K by employing (8)				
Method	**Features**	**Parameter**	PF	**Segments**	PF **(w/P)**	**Segments (w/P)**	**Parameters**	PF	**Segments**	PF **(w/P)**	**Segments (w/P)**
RRSC	MFCCs	$\lambda = 0.3$	0.54	37.1	0.54	13.5	$\lambda = 0.3, \tau = 0.70$	0.52	40.9	0.52	13.6
	Chroma	$\lambda = 0.1$	**0.47**	36.7	**0.53**	12.0	$\lambda = 0.1, \tau = 0.65$	**0.48**	28.3	0.51	12.8
	ATMs	$\lambda = 0.1$	**0.61**	6.1	**0.64**	6.1	$\lambda = 0.1, \tau = 0.11$	**0.59**	6.5	**0.60**	8.5
SSC	MFCCs	$\lambda = 0.7$	**0.55**	22.9	**0.57**	19.0	$\lambda = 0.7, \tau = 0.23$	**0.55**	19.4	**0.57**	5.8
	Chroma	$\lambda = 0.7$	0.47	33.8	0.50	21.6	$\lambda = 0.7, \tau = 0.19$	0.46	33.1	**0.54**	5.9
	ATMs	$\lambda = 0.3$	0.60	6.6	0.62	8.0	$\lambda = 0.3, \tau = 0.01$	0.59	6.1	0.54	3.6
LRR	MFCCs	$\lambda = 1.1$	0.53	38.2	0.53	13.4	$\lambda = 1.1, \tau = 0.65$	0.51	33.4	0.54	8.9
	Chroma	$\lambda = 0.5$	0.46	47.9	0.52	17.2	$\lambda = 0.5, \tau = 0.65$	0.47	43.5	0.52	11.9
	ATMs	$\lambda = 0.9$	0.59	17.6	0.60	7.6	$\lambda = 0.9, \tau = 0.13$	0.56	16.3	0.57	3.3
SDM	MFCCs	-	0.42	175.5	0.49	7.3					
	Chroma	-	0.43	150.4	0.49	24.5					
	ATMs	-	0.33	406.1	0.47	7.1					
[17]	Combination of MFCCs and Chroma	N/A	**0.63**								
[3]	Combination of MFCCs and Chroma	N/A	0.62								
[8]	Chroma	N/A	0.60								
[7]	MFCCs	N/A	0.60								
[6]	ATMs	N/A	0.59								

Table 1. Segment-type labeling performance on the Beatles dataset. In the last five rows the segment-type labeling performance on the Beatles dataset obtained by state-of-the-art methods with fixed $K = 4$ is shown.

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] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. 11th Int. Conf. Music Information Retrieval*, Utrecht, The Netherlands, 2010, pp. 625–636.

[2] L. Barrington, A. Chan, and G. Lanckriet, "Modeling music as a dynamic texture," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 602–612, 2010.

[3] F. Kaiser and T. Sikora, "Music structure discovery in popular music using non-negative matrix factorization," in *Proc. 11th Int. Conf. Music Information Retrieval*, Utrecht, The Netherlands, 2010, pp. 429–434.

[4] M. Bruderer, M. McKinney, and A. Kohlrausch, "Structural boundary perception in popular music," in *Proc. 7th Int. Conf. Music Information Retrieval*, Victoria, Canada, 2006, pp. 198–201.

[5] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2008.

[6] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "l1-graph based music structure analysis," in *Proc. 12th Int. Conf. Music Information Retrieval*, Miami, USA, 2011, pp. 495–500.

[7] 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*, vol. 17, no. 6, pp. 1159–1170, 2009.

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

[9] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[10] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 2790–2797.

[11] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2011, arXiv:1010.2955v4 (preprint).

[12] 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*, vol. 18, no. 3, pp. 576–588, 2010.

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

[14] D. Ellis, "Beat tracking by dynamic programming," *J. New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[16] R. Bhatia and F. Kittaneh, "Norm inequalities for partitioned operators and an application," *Math. Ann.*, vol. 287, no. 1, pp. 719726, 1990.

[17] R. Chen and L. Ming, "Music structural segmentation by combining harmonic and timbral information," in *Proc. 12th Int. Conf. Music Information Retrieval*, Miami, USA, 2011, pp. 477–482.