MUSICAL AUDIO SEMANTIC SEGMENTATION EXPLOITING ANALYSIS OF PROMINENT SPECTRAL ENERGY PEAKS AND MULTI-FEATURE REFINEMENT

P. Romano, G. Prandi, A. Sarti, S. Tubaro

Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

ABSTRACT

In this paper we present a novel hierarchical and scalable three-stage algorithm to effectively perform musical audio semantic segmentation. In the first stage, the energy spectrum of the entire audio track is analyzed to find significant energy textures that may characterize different semantic segments; in the second and third stages, tonal and timbric features are used to refine the segmentation by moving or deleting segment boundaries. Experimental results on a set of 58 songs show that our algorithm is able to attain good semantic segmentation just after the first step, with a precision of 64% and a recall of 96%. After second step the precision increases to 79%; the best precision result is obtained after the third step, where a value of 85% is reached. In this step the minimum average recall value of 92% is obtained.

Index Terms— Semantic Music Segmentation, Audio Structural Analysis, Audio Novelty Analysis.

1. INTRODUCTION AND RELATED WORK

Usually, when we listen to a piece of music we are able to detect and recognize a set of semantic parts or segments that constitute the high-level structure of the piece: for example, we can often easily detect an introduction, one or more verses, and the chorus. The problem of automatically detecting semantic segments is a challenging field of research for its intrinsic usefulness in practical Music Information Retrieval (MIR) applications: for example, a segment which is discovered as semantically meaningful may be extracted and used to give a short sample of the musical piece to the listener [1]; semantic segmentation may be exploited to give to the listener the possibility to navigate into the piece in a semantic way; discovering the semantic structure may be also a point of strength for music similarity computation and music recommendation applications [2]; also the interaction and the synchronization of the audio stream with other events may benefit from semantic segmentation.

Various approaches have been used in the literature to perform semantic segmentation; many works exploit a similarity measure between intra-song fragments to detect music track structure. The similarity measure may be visualized using a *similarity* or *distance matrix* [3]. In [3], Foote uses the similarity matrix to visualize the time structure of music and audio. In [4] the same author describes a novelty audio measure computed from the similarity matrix, which can be used to segment the musical piece. Bartsch and Wakefield [5] describe a method to find choruses by analyzing parallel high-correlation paths in the similarity matrix built using chroma-based audio representations. The most recurring chorus is used as a thumbnail of the musical piece. Ong and Herrera [6] use 2D morphological operators on similarity matrix to better recognize the fundamental structural elements used to describe the semantic global structure of the piece. Other works make use of machine learning techniques to segment a song; for example, Aucouturier and Sandler [7] present a segmentation algorithm based on a hidden Markov model; each state of the HMM, after appropriate training, corresponds to a different audio texture. Logan and Chu [8] propose a technique for extracting the 'key phrase' of a piece of music: after modeling the song using audio features, the structure of the song is discovered by training a HMM. The key phrase is then extracted using a heuristic approach. Exploiting the Extended Baum-Welch transformations, Sainath et al. [9] describe a segmentation method which identifies the most significant spectral changes to detect segment boundaries. Other works [10][11] exploit singular value analysis to cluster similar audio data.

In general, the segmentation algorithms presented in the literature are monolithic. The idea of structuring a segmentation algorithm in different steps has been used, for example, by Ong and Herrera [6], but, in general, without a specific and clear scalability goal. For our MIR applications, we needed an algorithm capable of good segmentation performance, but also scalable, i.e. composed of different stages of increasing complexity which can be used in different combinations to reach different complexity and performance results. Depending on the level of the needed segmentation detail, the algorithm presented in this article can be either stopped at the end of the third step or at intermediate step, achieving a good scalability of performance. To perform segmentation in the first step, we present a new segmentation algorithm based on spectral energy texture change detection. Furthermore, a combination of well-known techniques based on multi-feature similarity analysis are used to perform a robust boundary repositioning and similarity detection for segment merging in the second and third steps.

The paper is structured as follows. In Section 2 we give a general overview of the proposed algorithm. Section 3 describes in detail the operations related to the extraction of the coarse segmentation based on the analysis of spectral energy peaks. In Section 4 the two phases related to segmentation refinement using multi-feature vectors are described. Algorithm performance results are presented in Section 5. Finally, Section 6 draws some concluding remarks.

2. ALGORITHM OVERVIEW

In the first phase we try to detect the rhythmic and melodic patterns that constitute specific spectral textures in the audio signal spectrum. Such textures can be analyzed to reveal prominent spectral energy peaks and, in particular, their duration intervals into the musical signal. Using only the spectral energy computed directly from the signal, the goal of this phase is to extrapolate a first segmentation of the musical track by detecting important changes in spectral texture over time. The output of this phase is composed of a set of segment boundaries that, if needed, will be refined and pruned in the next phases.

In the second phase, timbral and tonal analysis is used to refine the boundary set extracted in the previous step. For each boundary,

a local novelty analysis is computed over a small time interval centered on the candidate boundary. During this phase a boundary can be repositioned more precisely or can be removed when one of the two related segments has been detected as too short.

The third phase performs segment merging exploiting the same timbral and tonal features of the previous step. Each segment is represented using a mean feature vector, and for each feature a segment-similarity matrix is computed. Two adjacent segments are merged together if they are detected as similar by all the segment-similarity matrices.

In the following sections we describe each phase in more detail.

3. SEGMENTATION THROUGH ANALYSIS OF SPECTRAL ENERGY PEAKS

Given a monaural audio signal $\mathbf{x}(n)$, sampled at 11025 Hz, the related spectrogram function $\mathbf{X}(w, f)$ is computed, where $w = 1, .., W$ is the analysis window index and $f = 1, .., F$ is the frequency bin index. A Hamming function is used to extract each signal window of length of 0.5 sec. The window overlap is set to 50%.

A gamma correction function is used over the spectrogram to highlight the spectral energy textures more clearly:

$$\hat{\mathbf{X}}(w, f) = \mathbf{X}^{\gamma}(w, f). \tag{1}$$

In our framework we use $\gamma = 0.3$. Based on $\hat{\mathbf{X}}(w, f)$ we select the prominent average-energy peaks. These peaks will be later used to select the most important spectral energy subbands to perform texture change detection. For each frequency bin the average energy over time is computed:

$$E_{avg}(f) = \frac{1}{W} \sum_{w=1}^{W} \hat{\mathbf{X}}(w, f). \tag{2}$$

Then, an adaptive threshold is applied to $E_{avg}(f)$. The threshold for the frequency bin f is defined as follows:

$$T(f) = \frac{E_{avg}(f-1) + E_{avg}(f) + E_{avg}(f+1)}{3} c_T. \tag{3}$$

We consider $E(0) = E(F+1) = 0$. The constant c_T depends on the spectrogram of the analyzed musical piece. Given the ratio r_T between the number of spectrogram coefficients above the $T(f)$ thresholds and the total number of spectrogram coefficients, the constant c_T is set to have $r_T = 0.35$. After thresholding, all the values $\hat{\mathbf{X}}(w, f) < T(f)$ are set to 0.

Given the function $E_{avg}(f)$, it is uniformly subdivided in three subbands, to obtain the functions $E_{avg,i}(f)$, $i = 1, 2, 3$. Let M be an operator which returns the local maxima of a function. M is applied one, two, or three times respectively on $E_{avg,1}$, $E_{avg,2}$. $E_{avg,3}$ to obtain three sets of local maxima:

$$\begin{aligned} m_1 &= M(E_{avg,1}(f)), \\ m_2 &= M(M(E_{avg,2}(f))), \\ m_3 &= M(M(M(E_{avg,3}(f)))). \end{aligned} \tag{4}$$

This approach is justified by the fact that the subband that contains the major part of the energy is the lowest one: all the local peaks associated to this subband are considered for the next step. Applying two or three times the M operator respectively on the second and third subband, allows to select only the most important peaks , by deleting maxima given by noise or by irrelevant, short audio signal components. The total number of selected prominent peaks is $P = |m_1| + |m_2| + |m_3|$.

Now, given a generic frequency bin f_p on which an energy peak $p \in m_1 \cup m_2 \cup m_3$ has been detected, we select a narrow energy subband around f_p and we compute the total energy $S_p(w)$ associated to such subband over time:

$$S_p(w) = \sum_{b=-B/2}^{B/2} \hat{\mathbf{X}}(w, f_p + b). \tag{5}$$

In our framework, we have chosen $B = 2$. The set of the extracted energy lines can be considered a reduced version of spectrogram, which will be used, after morphological filtering [12], to detect changes in spectral textures. Morphological filtering is applied to each function $S_p(w)$ to emphasize energy steps. A similar technique is used in [6] to emphasize the high-level structure of the similarity matrix. We process each energy function using opening and closing operators, as follows:

$$\hat{S}_p(w) = \gamma_{7.5}(\phi_{3.5}(S_p(w)) + \phi_{7.5}(S_p(w)) + \phi_{25}(S_p(w))). \tag{6}$$

The subscripts on opening operator γ and on closing operators ϕ indicate structuring elements of length respectively 3.5, 7.5, and 25 sec. Through morphological processing, algorithm tries to fill local energy holes in energy textures. Different structuring element lengths in closing filtering attempt to take into account different sizes of texture, while applying opening filter on the sum of closed energy functions removes filled areas with duration less than 7.5 sec.

To detect raising and falling points of each energy function $\hat{S}_p(w)$, the first derivatives $\hat{S}'_p(w)$ are computed. For each derivative a new function $H_p(w)$ is computed as follows:

1. The functions $\left|\hat{S}'_p(w)\right|$ and $H_p(w)$ are equally subdivided into non-overlapped windows Q_p of 7.5 sec.
2. For each window Q_p in $H_p(w)$, a single non-zero element is set at the centroid of the corresponding window in $\left|\hat{S}'_p(w)\right|$, with a magnitude equals to the area of $\left|\hat{S}'_p(Q_p)\right|$.

The functions $H_p(w)$ contain isolated peaks that allows to better detect the locations of slope changes in $S_p(w)$. A segmentation function $R(w)$ can be defined by summing together the $H_p(w)$ functions as follows:

$$R(w) = \sum_{P} H_p(w) \cdot c_p, \tag{7}$$

where $c_p = E_{avg}(f_p)$. The function $R(w)$ is then clipped by cutting values greater than a threshold r_c defined as follows:

$$r_c = \text{mean}(R(w)) + \frac{\max(R(w)) - \text{mean}(R(w))}{3}. \tag{8}$$

We call $\tilde{R}(w)$ the clipped $R(w)$ function. Now, finding the segment boundaries is a simple matter of selecting the peaks in the segmentation function $\tilde{R}(w)$. The approach used here is based on the following steps:

1. The mean value of $\tilde{R}(w)$, called $\tilde{R}_{avg}$, is used as threshold to generate two other functions, $A(w)$ and $U(w)$, defined as follows:

$$A(w) = \begin{cases} \tilde{R}(w) & \text{if } \tilde{R}(w) > \tilde{R}_{avg} \\ 0 & \text{if } \tilde{R}(w) \leq \tilde{R}_{avg} \end{cases} \tag{9}$$

$$U(w) = \begin{cases} \tilde{R}(w) & \text{if } \tilde{R}(w) < \tilde{R}_{avg} \\ 0 & \text{if } \tilde{R}(w) \geq \tilde{R}_{avg} \end{cases} \tag{10}$$

2. $A(w)$ and $U(w)$ are divided into non-overlapped windows of 3.5 sec. For each window Q the position w_z^A and w_z^U of two boundary candidates are computed respectively from $A(w)$ and $U(w)$ as follows:

$$w_z^A = \frac{\sum_Q A(w) \cdot w}{\sum_Q A(w)}, \quad w_z^U = \frac{\sum_Q U(w) \cdot w}{\sum_Q U(w)} \tag{11}$$

A boundary score corresponding to the sum of values of the related $A(w)$ or $U(w)$ function into the corresponding window Q is given to each boundary candidate.

3. All the boundary candidates computed using the function $A(w)$ are taken.
4. Candidate boundaries computed from $U(w)$ are taken only if they are at a distance greater than 4 secs from all the candidate boundaries computed from $A(w)$, and, at the same time, if their score values are greater than $\tilde{R}_{avg}$ threshold.

At the end, we have a set Z of boundaries which gives a first segmentation of the musical audio track.

4. SEGMENTATION REFINEMENT THROUGH TIMBRIC AND TONAL ANALYSIS

The first phase returns a coarse segmentation that presents imprecisions in locating boundaries. Moreover, in the first phase we use an approach which tends to over-segment the music track: this choice has been taken to avoid to irretrievably skip important semantic segments.

The problem of imprecise positioning of boundaries depends on the structure of the spectral energy texture, that may not have clear time limits, and on the morphological filtering, that tends to shift such limits due to opening and closing operations; moreover, the trajectories do not raise or fall like a step function but with a variable slope in different subbands, indeed the first derivatives return multiple values around structural changes.

The problem of over-segmentation is solved by deleting short segments and by merging adjacent similar semantic segments, in terms of timbric and tonal characteristics.

The timbric and tonal features used in our framework are listed in Table 1. For the computation of Audio Spectral Envelope coefficients, we have used a frequency band defined between 55 Hz and 5000 Hz, with a resolution of 0.25 octaves. MFCCs have been computed using 36 Mel filters.

Feature Type	Features	Ref.
Spectral	25 Audio Spectral Envelope coefficients	[13]
Perceptual	first 36 MFCCs	[13]
Tonal	12-bin chromagram	[5]

Table 1: Audio features used in our framework to perform boundary repositioning and segment merging.

4.1. Local novelty analysis for boundary repositioning, and elimination of short segments

For each boundary $z \in Z$ at position w_z we consider a 14 sec. rectangular window centered on the boundary. We subdivide the window into 0.5 sec frames, with an overlap of 50%. For each frame, a set of timbric and tonal features, as given in Table 1, is extracted. At the end of the extraction process, a matrix V_z of feature vectors $v_{z,k}$ is available for each boundary. Then, using the approach described in [4] a self-similarity matrix and a novelty function is computed for each window. The self-similarity matrix measures the similarity between each couple of feature vectors $(v_{z,j}, v_{z,k})$ using the cosine distance defined as follows:

$$D_C(j,k) = \frac{v_{z,j} \cdot v_{z,k}}{\|v_{z,j}\| \, \|v_{z,k}\|}. \tag{12}$$

The audio novelty function is computed correlating a Gaussian checkerboard kernel matrix along the main diagonal of the self-similarity matrix. As in [4], we use a smooth checkerboard kernel built using a radially-symmetric Gaussian function to avoid edge effects. The kernel size is 64×64, and, to build the smoothing Gaussian function, we used $\delta = 24$.
Each boundary is repositioned into the related window where the global maximum of the novelty function is detected.

In this phase we delete also segments shorter than 5.5 sec. To choose the boundary to delete (the left or the right boundary of the short segment), we compare the values of the scores assigned to the boundaries: the boundary with the lower score is deleted.

At the end, we have a refined set $\hat{Z}$ of boundaries.

4.2. Inter-segment similarity analysis for final segment merging

In the third phase we make use of tonal and timbric similarity to merge similar adjacent segments. The segment comparison is performed by computing self-similarity matrices over per-segment average feature vectors. In particular, for each segment, an average feature vector is computed; then, the average vector is decomposed in three parts corresponding to the three different feature spaces (Audio Spectral Envelope, MFCC, Chromagram). For each part, a self-similarity matrix is computed. At the end of the process we have three self-similarity matrices D_{ASE} (Audio Spectral Envelope domain), D_{MFCC} (MFCC domain), D_{Chroma} (Chromagram domain) that give information about intra-segment similarity using the three feature spaces separately. Based on work in [11] we use the larger terms in the SVD-based decomposition of such intra-segment similarity matrices to define clusters of similar segments. In our approach, we use this type of analysis to determinate the number of segment clusters. To evaluate the C larger terms in the SVD-based decomposition we use the following formula, computed on singular values λ_i of the diagonal matrix Λ:

$$I(C) = \sum_{i=1}^{C} \lambda_i / \sum_{j=1}^{|\hat{Z}|} \lambda_j. \tag{13}$$

$\left|\hat{Z}\right|$ is the total number of segments given from the second phase (and thus, it is the total number of elements on the main diagonal of Λ). We estimate the number of singular values C needed to make the value of $I(C) \approx 80\%$. Knowing C, which is the number of segment clusters we are considering, we select the greater $\left|\hat{Z}\right| - C$ local maxima in the actual inter-segment similarity matrix: this allow us to identificate couples of similar segments. Interconnected couples (i.e. that have a common segment) are clustered together. In particular, if two consecutive segments are positioned in the same cluster for all the three different feature domains, the boundary between them is removed.

At the end of the third phase, the final set of boundaries $\bar{Z}$ is returned.

5. EXPERIMENTAL RESULTS

We have conducted experimental tests on a heterogeneous audio database which consist of 58 songs from different music genres (rock, pop, world music, hard rock, punk, metal, ballad, electronic, hip hop). Each song has been converted from its original format to a 11025 Hz, 8 bit, mono-channel audio track. Results are evaluated using a ground truth segmentation, manually generated. For boundary evaluation we adopt the two metrics of *Precision* and *Recall* respectively computed as:

$$Precision = \frac{\#\ correctly\ detected\ boundaries}{\#\ total\ detected\ boundaries} \quad (14)$$

$$Recall = \frac{\#\ correctly\ detected\ boundaries}{\#\ ground\ truth\ boundaries} \quad (15)$$

To evaluate the correctness for boundary positioning, we consider a tolerance deviation of ± 2 sec from the ground truth boundary. As

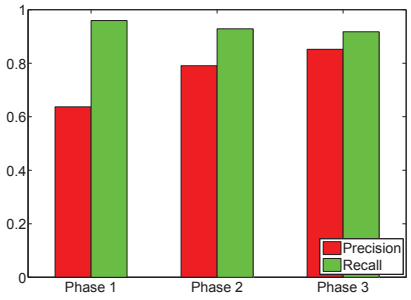

Fig. 1: Precision and recall results for the three phases

required, the first phase returns a over-segmentation in order to work with a set of candidate boundaries in which is present the subset of ground truth boundaries. As shown in Figure 1, this leads to a relatively low level of precision equal to 64%, but allows to work with a recall of 96%. From our results, the worst case of song recall is 83% highlighting the good ability of the first phase to detect structural changes. For the second phase, comparing the total positioning error of the whole set of boundaries, before and after relocation of local novelty, we observed an average error decrease of 88%. Also due to cancellation of short segments, the precision increases to 79%. The recall, instead, decreases to 93%. This means that a small number of true semantic boundaries are deleted. The segment merging approach adopted in the third phase allows to reach a precision result of 85%, while the recall decreases a bit and reaches a percentage of 92%. We have also conducted tests using lower time tolerances for boundary positioning evaluation. As shown in Figure 2, we observe modest reduction of precision and recall. With tolerance of 0.5 sec, for the last phase we obtain a precision score of 75% and a recall of 80%.

6. CONCLUSIONS AND FUTURE WORKS

In this paper a novel three-stage algorithm to effectively perform musical audio semantic segmentation has been proposed. In the first stage only the energy spectrum is analyzed to find changes in spectral energy texture, hence to detect a first set of semantic segments. In the second and third stages a multi-feature approach allows to refine the segmentation by moving or deleting segment boundaries.

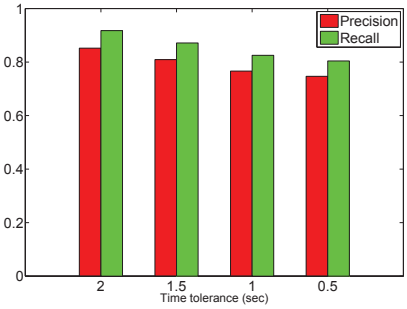

Fig. 2: Precision and recall values on the third phase, using different time tolerances to evaluate the correctness of boundary positioning.

Our tests show good results of precision and recall for all the three phases. Our future work will focus on using data computed in the third phase to perform semantic segment labeling, i.e. to give to segments a semantic name (chorus, verse, intro...). It is also our intention to experiment the proposed algorithm in a real-world application related to music recommendation field.

7. REFERENCES

[1] Wei Chai, "Semantic segmentation and summarization of music: methods based on tonality and recurrent structure," *IEEE Signal Processing Magazine*, vol. 32, pp. 124–132, March 2006.

[2] A. Bozzon, G. Prandi, G. Valenzise, and M. Tagliasacchi, "A Music Recommendation System Based on Semantic Audio Segments Similarity," *Proceedings of the IASTED International Conference*, vol. 612, no. 011, pp. 182.

[3] J. Foote, "Visualizing music and audio using self-similarity," *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pp. 77–80, 1999.

[4] J. Foote, "Automatic audio segmentation using a measure of audio novelty," *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 1, 2000.

[5] M.A. Bartsch and G.H. Wakefield, "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing," *Ann Arbor*, vol. 1001, pp. 48109–2110.

[6] B. Ong and P. Herrera, "Semantic segmentation of music audio contents," *Proceeding of International Computer Music Conference*, 2005.

[7] J.J. Aucouturier and M. Sandler, "Segmentation of Musical Signals Using Hidden Markov Models," *Preprints-Audio Engineering Society*, 2001.

[8] B. Logan and S. Chu, "Music summarization using key phrases," *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 2, 2000.

[9] T. Sainath, D. Kanevsky, and G. Iyengar, "Unsupervised Audio Segmentation using Extended Baum-Welch Transformations," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[10] S. Dubnov and T. Apel, "Audio Segmentation by Singular Value Clustering," *proceedings of International Computer Music Conference (ICMC), Miami*, 2004.

[11] J.T. Foote and M.L. Cooper, "Media segmentation using self-similarity decomposition," *Proceedings of SPIE*, vol. 5021, pp. 167–175, 2003.

[12] M. Van Droogenbroeck and MJ Buckley, "Morphological Erosions and Openings: Fast Algorithms Based on Anchors," *Journal of Mathematical Imaging and Vision*, vol. 22, no. 2, pp. 121–142, 2005.

[13] H.G. Kim, N. Moreau, and T. Sikora, *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*, Wiley, 2005.