Music Information Retrieval
Developing Tools For Musical Content Segmentation and Comparison

Ricardo Santos
Instituto Superior Técnico, Universidade Técnica de Lisboa,
Av. Rovisco Pais, 1049-001 Lisbon, Portugal,
rjsds@ist.utl.pt

Abstract. Music Information Retrieval (MIR) is now well established and comprises many techniques that have matured enough to be used in a context of a toolkit that allows the automation of common tasks in this discipline. The objective of this work was the implementation of tools in C++ to detect audio segment borders and repetitions, as well as to determine music distance measures that are consistent with human perceptions of music. A consequence of this process was the creation of a considerable code base, which can leverage future improvements and additions.

Among the most relevant, an implementation was made of a Gaussian checker kernel filter for music on-set detection and a two-stage K-Means clustering-based approach was used to identify segments that repeat themselves in songs. A clustering-based approach to music distance calculation, that relies on the Earth Mover’s Distance (EMD), was also implemented. Several distance measures were developed to compare clusters, including the Kullback-Liebler divergence of cluster points modelled with one of two different probability density functions that were implemented.

The music on-set detection implementation obtained an F measure of 62% which is comparable to the results obtained in reference works, and it was shown that the automatic segmentation retrieved more than 71% of segments determined to be similar by one of the implemented distance measures. Positive results where obtained for a small scale evaluation of the distance measures developed, although more extensive experimenting with these tools should be considered in future work.

1 Introduction and Objectives

This work describes a preliminary effort to develop a toolkit written in C++ for common tasks in music retrieval, aiming at the integration of a music Content Recommendation System (CRS). It covers two common tasks in this field, with a broad range of applications: music segmentation into humanly identifiable musical segments and music similarity measures. Over the next sections it will be referred to as the Shkr project.
1.1 Segmentation
Focus was given to detecting the humanly perceived portions of popular music that are commonly mentioned as “choruses”, “verses”, “intros”, “bridges” or “breaks”, and are usually repeated (sometimes with variations). This specific segmentation task was approached by several researchers (refer to Section 2), although the range of musical genres to which it is applicable is reduced, since it does not typically occur in all of them. Only Pop and Rock music was used in the segmentation evaluation corpus, in order to limit the scope of the evaluation to genres where this type of segmentation is usually applicable.

Target applications of this type of segmentation are, for example, “Jump to the next section” features in media players, which can allow a first time listener to quickly browse a song by listening only to the portions of it that are significantly different, and “Audio thumbnailing”, in which segmentation can be used to provide a list of candidate audio excerpts, that can be presented to a listener for fast musical identification or evaluation.

1.2 Distance Measure
In the scope of this work, music distance was approached as a tool to produce a measure by which an ordered playlist of similar songs in a corpus can be determined. The applications of such tools range from the suggestion of similar music to a listener, based on the musical signal instead of meta-data tags, and playlist generation/sorting (e.g. emphSchnitzer [1] to query by humming (e.g. Midomi [2]).

1.3 Objectives Overview
- To be able to detect humanly perceived musical events, such as the occurrence of choruses, intros, verses, vocal and non-vocal segments, or even musical notes and chords.
- To be able to quantify similarity of musical content
- To compile a structured and documented toolkit with the necessary tools for fulfilling the objectives above.

2 Strategies

2.1 Feature Extraction and Parametrization
*Mel Frequency Cepstral Coefficients* MFCCs have been used with positive results in several different MIR related tasks, from segmentation and structure finding (e.g. Levy et al. [6]), to music similarity measurement (e.g. Logan et al. [9]). They were the main feature used in the experiments made with the implemented algorithms. 41 MFCC Coefficients per frame where extracted, although the first, that contained the signal, energy was discarded. The extraction was made using short time analysis, using (Hanning filtered) windows with a duration of 200ms, separated by 100ms intervals.
2.2 Segmentation

Over the next sections, the term “segment border” is used when referring to points in a musical piece timeline, where musical events and changes (also referred to as on-sets) occur. “Segments” refer to spans of time in the music that are encompassed between “borders”, and “labels” are tags given to segments, which allow the grouping of similar or repeated “segments”.

2.2.1 Border Extraction from Autocorrelation Matrices

A widely used method to determine the boundaries of “events” in music, that was introduced by Foote [3]. It is a strategy for finding on-sets in audio where a significant change occurs. In the original work, Short Time Fourier Transform (STFT) coefficients were used as features, but authors such as Ong [4] and Paulus et al. [5] have used features other features, such as MFCCs, to produce the correlation matrices in which this process relies on.

The automatic identification of the visible pattern transitions in the correlation matrices can be done by calculating the correlation of the distance matrix with a smaller Gaussian checker kernel matrix, along the distance matrix’s diagonal as described in by Foote [3]. This process produces a vector, the maxima of which correspond to the pattern transitions seen on the distance matrix and therefore, the musical events borders. The resolution of the identified transitions varies significantly depending on the system’s parametrization.

Additionally, a moving average filter with a range of 40 frames is used on the retrieved borders, to improve the accuracy of the results. The peaks of the resulting vector constitute the resulting set of borders matched.

2.2.2 Clustering-Based Segmentation

The cluster-based segmentation approach used in this work is based on the one proposed by Levy et al. [6], of which the implementation used in the Segmentation plug-in freely available for Sonic Vizualizer [7]. In that work, a hidden Markov model (HMM) with 40 states was trained with 21-dimensional AudioSpectrumProjection [8] feature vectors extracted from the song. The features are subsequently Viterbi-decoded using the trained model and the sequence of most likely state assignments is retrieved for each beat of the song. This results in the sequence of “low-level” labels which in this work will be referred to as L1 - Level 1 - labels (or clusters).

In Levy et al. [6] it is further shown that the “high-level” labels can be extracted by using histograms of state assignments, taken from a sliding window of beats, and then using K-Means clustering with a lower number of clusters (as much as the desired number of “high-level” labels), to group the resulting histograms. The “high-level” labels are referred to as L2 (Level 2) labels or clusters in this work.
From experimentation and prototyping, it was observed that a similar L1 sequence of cluster assignments could emerge by using K-Means clustering directly on the features extracted from the music signal. The remaining portion of the algorithm is therefore similar to that of Levy et al. [6].

2.3 Distance Measures

The concept for distance measure strategy used departed mainly from the work of Logan et al. [9], in which song signatures are obtained from (K-Means) clustering (using 16 clusters) of 12, 19 and 29 dimensional MFCC feature vectors. The signatures were composed of each cluster’s mean, covariance, and weight. The song distance was measured was obtained from the cluster signatures using the Earth Mover’s distance (EMD), which is used to determine the minimum “work” of transforming one signature into another.

In this work, a Gaussian multivariate PDF and and the average of the Kullback-Liebler divergence of each feature modelled as a Gaussian univariate PDF was used instead, for for comparing clusters. Additionally, an Euclidean distance was also used for the same purpose by comparing their centres, and a covariance inter-cluster distance was also implemented for control purposes.

3 Architecture

3.1 Overview

While all of the segmentation and distance measure algorithms are part of the C++ implemented code base, several external tools are necessary to provide functionality such as configuration, audio conversion and feature extraction.

Audio conversion from MP3, Ogg [10] or Flac [11] formats was performed using Sox [12]. The MFCC extractor of the Hidden Markov Model Toolkit [13] was used to produce features that are used by Shkr.

On the output end of the chain, song distance measures, segmentation evaluation statistics, and other relevant information to the tasks performed are sent to the standard output of the terminal, within which Shkr must be run. Ground truth annotation is read from a SegmXML format as described in Peiszer [14], and graphics output can be used to export the contents of a Matrix data structure into Portable Anymap Format (PNM) of 8bit or 16bit, monochromatic or RGB (Red,Green,Blue) formats.

3.2 Processing Chains

Most of the strategies used in this work require several stages of processing where different algorithms manipulate and build on the output produced by previous stages. In audio editing and processing it is common to define processing chains, where each step either changes the data produced by the previous one, or produces additional data. These ideas are developed in Tzanetakis [15], where
a new model for creating implicit connections between components that require them is presented as it was applied in the Marsyas [16].

In Shkr processing chains were modelled using a decorator pattern as described in Eckel [17], since it suitable for modelling chains of processing steps that modify or add to the data being decorated. A Matrix object, containing data that needs to be passed on to the next stage in the chain, is used as the default decorated object, but the results of each processing stage can always be retrieved at any time during the chain execution or after the chain is concluded by use the identification tags passed onto each chain process. The ability to connect processing stages regardless of their position in the chain and the data types exchanged can be seen as a form of patching similar to those described by Tzanetakis [15].

3.3 Song Data Modelling and Comparison

The concept used for modelling song data in the context of song comparison was somewhat different to that used in segmentation. The objective was to be able to perform song comparison with something as simple as \( \text{distance} = \| \text{Song}A - \text{Song}B \| \). The SongA object should contain within itself the mechanisms to allow it to be compared with Song object B. The same principle was also adopted for Cluster objects containing the points of each cluster, its centre and statistic measures associated with it have also the mechanisms that allow them to determine the distance from other clusters.

4 Evaluation

The corpus of 61 Pop and Rock songs used in the segmentation evaluation is a sub-set of the Paulus and Klapuri annotated corpus [5], although the annotations used were obtained from the corpus converted by Peiszer [14], which are available at [18].

A total border matching tolerance of 4 seconds was used, since the exact location of a segment border might not be easily determined, and manual annotations suffer from inaccuracy that was empirically determined to be of up to 2 seconds in either direction of the song timeline. However, each ground truth border was only matched once to avoid false positives.

The corpus used for song similarity is a selection of 25 songs, organized into 5 groups, each group with 5 songs. The musical pieces within each group have the same author(s), album, genre, and the same predominant musical instruments. An attempt was made to create groups as distinct as possible among themselves in what concerns timbre, which MFCCs are documented to discriminate.

4.1 Song Segment Distance Measure

For both ground truth and automatically identified segments, the average distance of segments with the same label was computed and compared with the
average distance of segments with a different labels. Whenever the inter label average distance was lower that the average distance of the segments to those of all other labels, a match was considered to be found. Finally, the recall was obtained for both ground truth segments and automatically retrieved segments. The diagonal of the inter-segment distance matrix was not taken into account when calculating the averages since the distance between an audio segment to itself is always 0. The distance measure used in this evaluation was the average of N symmetric KL divergence distances between N Gaussian PDFs, parametrized from the N-dimensional feature vector belonging to a segment.

4.2 Success Metric for Song Distance

In order to determine a success rate of each distance measure, for each song in the corpus, the top 5 closest songs according to the distance measures are retrieved. Of those 5 songs, for each that belongs to the category of the first song a score point is added. An overall score is then calculated per category and for the full corpus, by dividing the sum of the partial scores from each category by the number of maximum points possible per category (25) and by the maximum number of points for the songs in the corpus (125), respectively.

5 Results and Conclusion

5.1 Result Summary and Discussion

In what concerns segmentation, the results can be summarized as follows:

- The border matching using a Gaussian Checker Kernel resulted in a maximum F measure of 62% (refer to Figure 1), which is within the range of the values obtained from previous works like Ong [4] and Peiszer [14] using the same algorithm.
- The border matching using a clustering-based segmentation approach resulted in low values of F measure of only 46% (refer to Figure 2). However, this is still higher that the results reported in Levy et al. [6] for experiments using similar algorithms.
- The evaluation of the segmentation through the use of a distance measure, was an attempt to establish a utility measure of the automatic segmentation produced by the cluster-based segmentation. With the criteria used, it was shown that while a manually annotated corpus contains approximately 88% of similar segments that are annotated with the same label, the cluster-based segmentation was able to identify segments that are between 71% and 76% similar among those with the same label (refer to Figure 3).
- Empirical listening of the music segmentations produced also proved that the results of the automatic segmentation were positive, in particular when the number of L2 clusters used was closer to the number of manually annotated labels in a song.
Fig. 1: Border matching performance measures for the *Gaussian Checker Kernel* approach. The Precision, recall and F measure percentage for all tested combinations of standard deviation of the *Gaussian Kernel* are depicted.

Fig. 2: Border matching precision and recall percentage, for cluster-based approach for the range of tested L2 cluster values. 80 L1 clusters and a 60 frame histogram sliding window size. The F measure is also depicted.

For the music distance measures implemented a presentation (refer to Figure 4) and discussion of the results follows:

- The Euclidean distance between clusters was the most successful measure employed. This indicates that the centre of feature space, defined by each category of songs evaluated, was itself a discriminator of the category, which was more evident in the single cluster experiments.
- In the task of finding the 5 closest songs from the biased corpus, experiments with multiple clusters obtained worse results than using just 1 cluster. This
Fig. 3: Average and total correct labels from GT and AS for the range of tested L2 clusters, using the the average of the KL divergence of per feature, where the points for each feature are modelled with a Gaussian PDF.

indicates that an increased number of clusters will not necessarily hold better results. The Euclidean cluster distance measure, seems to counter this tendency, since for 20 clusters the success score is 6.4% better that with 10 clusters, although it is still lower than the score obtained with just 1 cluster.

• The covariance cluster distance measure obtained consistently bad results, since the average (centre of the cluster) is subtracted in it’s calculus. This is coherent with the conclusions above.

• The average KL divergence of a single Gaussian PDF per feature distance measure, when employed to evaluate similarity among segments supposedly similar according to manual annotations, proved to return consistent results both empirically, by identifying incorrectly manually annotated segments, and in the context of the evaluation of the automatic segmentation by comparison with the ground truth segmentation.

• The KL divergence using a multivariate Gaussian as a PDF for modelling the clusters held positive results, but worse than those obtained by modelling each feature as a Gaussian independently. This indicates that a combination of the later might be a more adequate PDF to model the feature space.

• The Earth Mover’s distance measure implementation was validated since as show, with the exception the covariance cluster distance measure, all others produced positive scores with more that 1 cluster, and in the case of the Euclidean cluster distance, 20 clusters held better results than 10 clusters, as expected.

• While the KL divergence based strategies had generally lower performance that the Euclidean distance, it is not clear that this advantage will hold in a more heterogeneous corpus, and for an higher number of clusters. However, it is clear that the Euclidean distance and KL based methods held positive results.
5.2 Future Work

From the author’s experience, future work and improvements to Shkr should be concentrated in:

- Moving the feature extraction inside the code base of Shkr, or at least under the control of the Shkr code base, to allow for the integration of beat and metric.
- Optimizing the available functionality by resorting to feature testing and selection, and possibly include PCA among the tools available to reduce the dimensionality of the features when necessary.
- Playlist retrieval and optimization strategies are desirable, and will greatly improve the usefulness of the implemented distance measures, specially in the context of a CRS.
- Other Classifiers such as HMM or SVN should be included as well, since K-Means clustering does not always converge, and results rely on random seeds, which can cause inconsistencies, specially song distance measures.

5.3 Objectives versus Challenges

It is the author’s opinion that these results validate the implementations, show the applicability of the strategies used, and therefore the objectives of this work were accomplished. Although there were several tasks being approached and few task-specific optimizations were introduced, the results were still comparable to those obtained from other authors. Shkr is still a developing project, and
while for segmentation tasks applicable results can already be obtained, for the
distance measures more research and testing is necessary, and hence the small
scale of the test performed in the scope of this thesis.

However, a considerable code base with numerous useful resources and a
flexible design that can support several tasks in MIR was built, and while the
effort was not invested in highly desirable improvements such as feature selection,
or beat detection integration, foundations were laid that can help leverage such
functionality in future.

References

1. Schnitzer, D. Mirage, high-performance music similarity computation and auto-
3. Foote, J. Automatic Audio Segmentation using a measure of audio novelty. pro-
4. Ong, B. S. Towards Automatic Music Structural Analysis: Identifying Character-
5. Paulus, J., Klapuri, A. Music Structure Analysis by Finding Repeated Parts. pro-
standards/mpeg-7/mpeg-7.htm (Visited on 2010/10/10).
2010/09/28).
on 2010/09/03).
14. Peiszer, E. Automatic Audio Segmentation: Segment Boundary and Structuzre
15. Tzanetakis, G., Bray, S. Implicit patching for dataflow-based audio analysis and
2005.
marsyas.info/ (Visited on 2010/09/03).
17. Eckel, B. Thinking in Patterns - Problem-Solving Techniques Using Java, Revision
mindview.net/Books/TIPatterns/ (Visited on 2010/09/03).