A MATLAB TOOLBOX TO COMPUTE MUSIC SIMILARITY FROM AUDIO

Elias Pampalk
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, A-1010 Vienna, Austria

ABSTRACT

A Matlab toolbox implementing music similarity measures for audio is presented. The implemented measures focus on aspects related to timbre and periodicities in the signal. This paper gives an overview of the implemented functions. In particular, the basics of the similarity measures are reviewed and some visualizations are discussed.

1. INTRODUCTION

New technologies are needed to access large digital music collections (e.g. automatic playlist generation, recommendation, query-by-example, hierarchical organization and visualization). One important building block is a computational model of music similarity based on audio analysis.

Although approaches to compute similarity have been published (e.g. [5, 6, 1, 7, 8, 4]) hardly any open source implementations are available. One of the few exceptions is the CLAM framework which offers functions to compute descriptors which can be used for music similarity. [1] Another exception is the Marsyas framework for computer audition [12] which implements several descriptors for genre classification.

In this paper an open source Matlab toolbox is presented wich is available on the Internet. [2] The toolbox implements several similarity measures and functions to visualize intermediate steps in the computations. Furthermore, some basic functionalities are included to create the islands of music metaphor where islands represent clusters of similar pieces [7]. Although the toolbox is not suited for large scale processing (i.e. collection sizes beyond 5000 pieces) it can serve as reference implementation and as playground to study effects of parameters. First experiments in this direction using parts of the current toolbox were reported in [9]. In this paper the main functions are presented and some demonstrations are given.

2. SIMILARITY MEASURES

There are four main issues involved in designing similarity measures. First it is necessary to decide which features to extract from the audio signal. This decision depends on the targeted concept (e.g. timbre, rhythm, harmony, melody etc.) and involves signal processing, psychoacoustics, and music perception. The second issue is closely related to the first and deals with how to summarize and describe a piece of music based on the extracted features. Often pieces are very inhomogeneous (e.g. *Bohemian Rhapsody* by *Queen*). Simply calculating the mean of the extracted features is usually not useful. The third issue is how to compare the representation of one piece with another. Depending on the previously chosen representation this might be difficult. The visualizations in the toolbox support the user in understanding the different choices for each decision. The fourth issue is computational efficiency. It is not possible to model every nerve cell in the human auditory system when processing music archives with terabytes of data.

In the following we briefly review the similarity measures and how they are implemented in the toolbox.

2.1. Frame Clustering and Cluster Model Similarity

The idea of FC/CMS was first published by Logan and Salomon [6] and improved and optimized by Aucouturier and Pachet [1, 2, 3]. A piece of music is cut into thousands of very short frames (of about 20-30ms length). Each of these frames is described in terms of loudness per frequency band. Usually Mel frequency cepstrum coefficients (MFCCs) are used, however any loudness/frequency model could be used (e.g. sone/bark).

Once this first step is computed there are two steps to compare the similarity between two pieces. First, in the frame clustering (FC) step, the frames are clustered into usually 3-40 groups (using either k-means or Gaussian mixture models (GMM) with expectation maximization) and the diagonal covariance of each group is computed.

Second, in the cluster model similarity (CMS) step, the similarity of two pieces modeled by clusters is computed. Two different approaches have been used. Logan and Salomon proposed the Earth Mover's Distance (EMD, originally developed as distance measure for image retrieval, see [10] for details) in combination with the Kullback-Leibler (KL) divergence to compute the distance between

[1] http://www.iua.upf.es/mtg/clam
[2] http://www.oefai.at/˜elias/ma

two clusters.

Aucouturier and Pachet proposed using Monte Carlo (MC) sampling to measure the similarities of the cluster models. In particular, given two pieces A and B a large sample (2000-5000) is drawn from each of the cluster models. Then the likelihood that the sample drawn from A was generated by model B and vice versa is computed and used as distance function. The advantage of EMD-KL compared to MC is that it is computationally much faster, however in an evaluation presented in [3] MC outperformed EMD-KL.

The computation of the frames is either done using “ma_mfcc” or “ma_sone”. In both cases the audio signal must be in the memory (usually loaded by “wavread”). FC/CMS is implemented by the functions “ma_fc” and “ma_cms”. In both cases parameters define which variation (i.e. k-means or GMM, EMD-KL or MC) to use. Any combination is possible. For the GMM and MC the Netlab[3] toolbox and for the EMD the code published by Yossi Rubner is required.

In the remainder of this paper “AP 30” is used to describe FC with 20 MFCC coefficients (ignoring the 0th coefficient) and a GMM with 30 centers. For CMS the MC is used with 2000 samples as suggested by Aucouturier and Pachet. “LS 30” is used to describe the same as AP 30 but using k-means instead of a GMM and EMD-KL instead of MC as suggested by Logan and Salomon.

2.2. Spectrum Histograms

The SHs [8] are a simple approach to summarize the spectral shape. They are based on a sone/bark representation of the audio signal. The SHs describe a piece by counting how many times each loudness level was exceeded in each frequency band. The distance between two SHs is measured with the Euclidean metric. SHs are computationally much faster than FC/CMS variations, however their performance is also significantly worse. The functions needed to compute SHs are “ma_sone” and “ma_sh”.

2.3. Periodicity Histograms

PHs were originally presented in the context of beat tracking [11]. Details of the differences between the PH similarity measure and previous approaches can be found in [8]. The idea is to describe periodically reoccurring beats. The features are extracted by further processing the sone/bark representation. The distance is measured using the Euclidean metric. Although the PH is not a good similarity measure it is a good starting point to compute higher level descriptors. The functions needed to compute PHs are “ma_sone” and “ma_ph”.

2.4. Fluctuation Pattern

FPs are an alternative approach to describe periodicities. The main difference between FPs [7] and PHs is that the FPs include information on the energy distribution in the frequency spectrum which the PHs discard. Another differences is that FPs use a FFT instead of a comb-filter to find periodicities in the critical-bands (bark-scale). Furthermore, while the PHs use a resonance model which has a maximum at about 120bpm the FPs use a fluctuation model which has a peak at 4Hz (240bpm). The distance is measured using the Euclidean metric. The functions needed to compute FPs are “ma_sone” and “ma_fp”.

	AP 30	LS 30	SH	PH	FP
Sone			6m	6m	6m
MFCC	7m	7m			
Features	132m	15m	48s	12m	23s
Distances	74m	4m	<1s	<1s	<1s
Total	213m	26m	7m	18m	6.5m
R-Precision	0.43	0.38	0.21	0.13	0.25

Table 1. Small Evaluation. Computation times (in minutes [m] and seconds [s]) and R-precision values.

3. EXAMPLES

In this section some examples of how the toolbox can be used for visualizations and evaluations are given. Further examples are included in the code of most functions and are executed when the functions are called without input arguments.

3.1. Similarity Measures

Figures 1-4 show how the representations used by the similarity measures reflect characteristics of the music. In all figures, darker shadings correspond to higher values. Each subplot represents a different piece of music, namely (a) Chopin - Waltz (Op.69 No.1), (b) Bon Jovi - Its My Life, and (c) Kai Tracid - Tiefenrausch. The later belongs to the genre dance/electronic and has very strong beats.

3.2. Simple Evaluation

There are different ways to evaluate similarity measures. A true evaluation can only be performed in the context of a specific application and usually requires extensive user studies. However, several work arounds to compare different parameter settings have been presented, e.g. in [3] the authors use the R-precision. A simple script which demonstrates how such an evaluation can be performed is included in the toolbox (“ma_simple_eval”).

Results from such an evaluation are shown in Table 1 including the computation times. The data set consists of 118 pieces classified into 19 categories. Note that the computation time for the feature computation for AP 30 can be reduced by using less iterations of the expectation maximization algorithm and stopping the iteration when the Gaussian mixture model converges. As can be seen the performance of AP 30 clearly outperforms SH, PH, and FP, while it is comparable to LS 30. Preliminary experiments indicated that the main difference in performance

[3] http://www.ncrg.aston.ac.uk/netlab

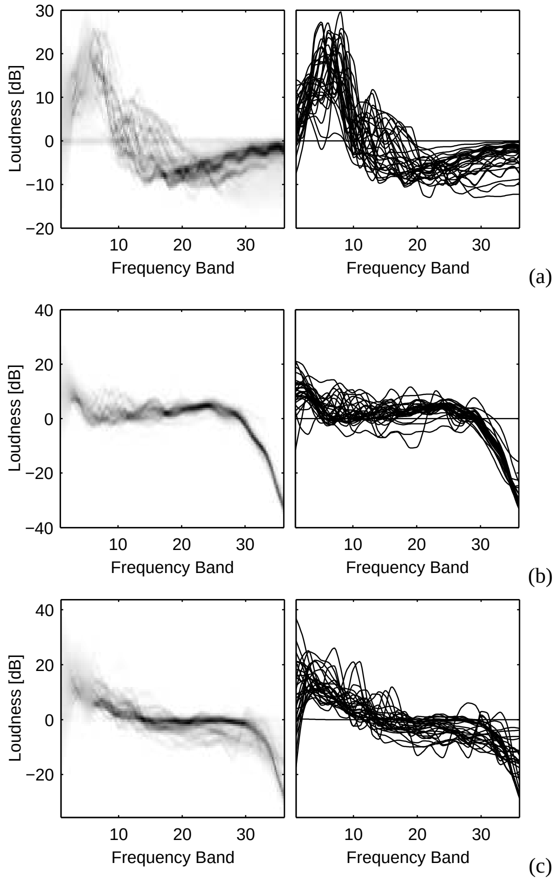

Figure 1. Frame Clustering (AP 30). On the left, the shadings correspond to the combined density distribution of the 30 Gaussians. On the right, the lines depict the cluster centers (i.e. the means of the Gaussians).

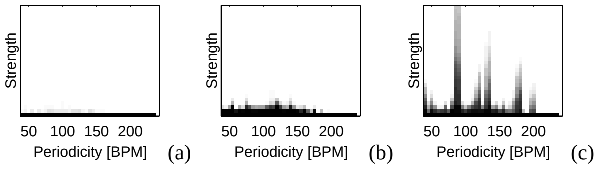

Figure 2. Periodicity Histograms.

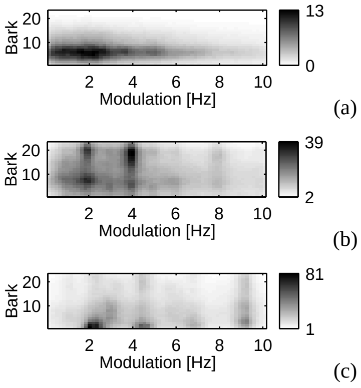

Figure 3. Fluctuation Patterns.

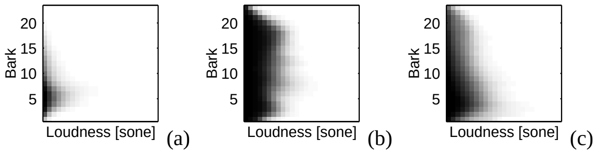

Figure 4. Spectrum Histograms.

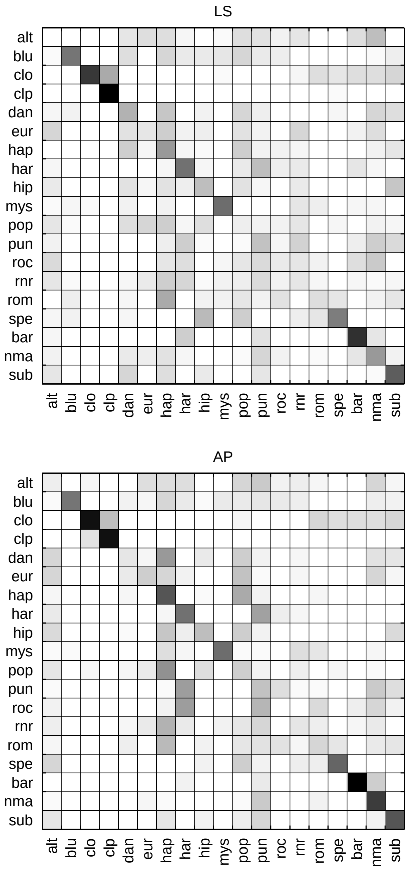

Figure 5. Confusion matrices for LS 30 and AP 30. The rows refer to the estimated classes and columns to the true classes. Black corresponds to 100%, white to zero.

between LS and AP is a result of using EMD instead of Monte Carlo sampling.

In addition to comparing R-precision values it is interesting to study confusion matrices. We compute the confusion between classes A (true) and B (estimated) as the average ratio (over all elements in A) of elements from B in the set of n nearest neighbors to an element from A. Where n is the number of elements in B. As can be seen some genres are very clearly distinguished such as blues (blu), classic orchestra (clo), classic piano (clp), and speech (spe). Furthermore, we had 3 classes which only contained pieces from one artist (bar, nma, sub), all of which are also well distinguishable. Most of the other classes are poorly distinguished, including alternative (alt), pop, and romantic dinner music (rom). One explanation is that the classes are defined subjectively and are not necessarily confined to distinctive timbres. As expected the main characteristics of the AP 30 and LS 30 confusion

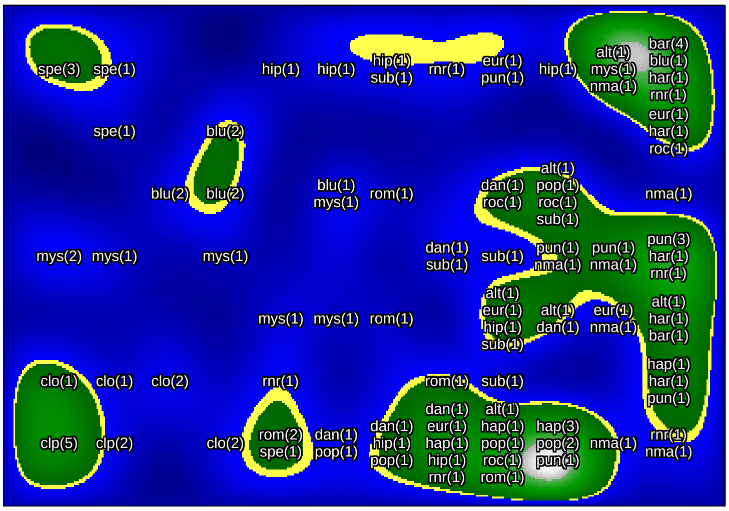

Figure 6. Islands of Music created using spline interpolation and SDH with $s = 2$.

matrices are very similar.

3.3. Islands of Music

In combination with the SOM[4] and SDH[5] toolboxes it is possible to create the Islands of Music visualization with only few lines of code.

An example is shown in Figure 6, example code can be found in "ma_simple_iom". The distances are computed using FC/CMS (AP 30) on the same data used for the small evaluation. In the upper left there is an island representing speech (spe), in the lower left there is an island with classic piano (clp) and classic orchestra (clo). To the south-east of the speech island there is an island with mainly blues (blu), to the north-east of the blues island there are some hip-hop pieces (hip). However, further to the east of the map the distinctions are not as clear. This is confirmed by the confusion matrices. For example, rock'n roll (rnr), rock (roc), and punk rock (pun) seem randomly distributed.

4. ACKNOWLEDGMENTS

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Education, Science, and Culture and by the Austrian Federal Ministry for Transport, Innovation, and Technology. This research was supported by the EU project FP6-507142 SIMAC[6] and the national project Y99-INF, sponsored by the Austrian Federal Ministry of Education, Science, and Culture in the form of a START Research Prize.

5. REFERENCES

[1] J.-J. Aucouturier and F. Pachet, "Music similarity measures: What's the use?," in *Proc of ISMIR*, 2002.

[2] J.-J. Aucouturier and F. Pachet, "Finding songs that sound the same," in *Proc of IEEE Benelux Workshop on Model Based Processing and Coding of Audio*, 2002.

[3] J.-J. Aucouturier and F. Pachet, "Improving Timbre Similarity: How high's the sky?," in *Journal of Negative Research Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[4] A. Berenzweig, D.P.W. Ellis, and S. Lawrence, "Anchor space for classification and similarity measurement of music," in *Proc of ICME*, 2003.

[5] J. T. Foote, "Content-based retrieval of music and audio," in *Proc of SPIE Multimedia Storage and Archiving Systems II*, 1997, vol. 3229.

[6] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc of ICME*, 2001.

[7] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," in *Proc of ACM Multimedia*, 2002.

[8] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," in *Proc of ISMIR*, 2003.

[9] E. Pampalk, S. Dixon, and G. Widmer, "On the evaluation of perceptual similarity measures for music," in *Proc of DAFx*, 2003.

[10] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Intl Journal of Computer Vision*, vol. 40, no. 2, 2000.

[11] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *JASA*, vol. 103, no. 1, 1998.

[12] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organised Sound*, vol. 4, no. 3, 2000.

[4] http://www.cis.hut.fi/projects/somtoolbox

[5] http://www.oefai.at/˜elias/sdh

[6] http://www.semanticaudio.org