A SIMILARITY MEASURE FOR AUDIO QUERY BY EXAMPLE BASED ON PERCEPTUAL CODING AND COMPRESSION

Marko Helén and Tuomas Virtanen

Tampere University of Technology
Institute of Signal Processing
Korkeakoulunkatu 1, FIN-33720 Tampere, Finland
marko.helen@tut.fi

ABSTRACT

Query by example for multimedia signals aims at automatic retrieval of samples from the media database similar to a user-provided example. This paper proposes a similarity measure for query by example of audio signals. The method first represents audio signals using perceptual audio coding and second estimates the similarity of two signals from the advantage gained by compressing the files together in comparison to compressing them individually. Signals which benefit most from compressing together are considered similar. The low bit rate perceptual audio coding preprocessing effectively retains perceptually important features while quantizing the signals so that identical codewords appear, allowing further inter-signal compression. The advantage of the proposed similarity measure is that it is parameter-free, thus it is easy to apply in wide range of tasks. Furthermore, users' expectations do not affect the results like they do in parameter-laden algorithms. A comparison was made against the other query by example methods and simulation results reveal that the proposed method gives competitive results against the other methods.

1. INTRODUCTION

The management of ever growing multimedia databases is very time consuming when done completely manually. This is why automatic systems are required to lighten the job. Query by example aims at automatical retrieval of samples from a database, which are similar to a user-provided example. For example, a user gives an example of a dog barking and the system returns all the samples from the database which contain dog barking.

The concept of similarity itself is very problematic. Measuring similarity of audio samples without annotations is very difficult comparing to a text-based search, since the similarity in signal level does not correlate to human's impression of similarity. For example in the situation when there is an example of male speech, it is impossible to know whether the user wants samples from the same speaker, or about the same topic.

Most of the existing audio query by example systems approach the problem as follows. first, features from the example signal and from the database signals are extracted. Second, the distance between the example signal and each database signal is estimated. Finally, the samples which have the shortest distance to the example are retrieved.

Pampalk estimated a Gaussian mixture model (GMM) for the example and estimated the similarity by the likelihood that the database sample was generated by this model [1]. Mandel and Ellis [2] calculated the mean of each feature over the whole sample and used the Mahalanobis distance between the samples as a similarity measure. They also used the Kullback-Leibler divergence between two GMMs to estimate the similarity.

Helén and Lahti [3] used a histogram based method, which generated feature histograms for each signal, and calculated the distances between these histograms. They also used a method, which generates hidden Markov model (HMM) for each sample and also a universal background model using the whole database. Then they estimate whether it is more likely for the database signal to be generated by the example HMM or the background model. Helén and Virtanen proposed a method for estimating similarity by calculating the Euclidean distance between two GMMs of the features [4].

When measuring the similarity between two samples, parameters like the feature set have to be decided a priori. The choice of these parameters is crucial for the results and choosing the right parameters requires a lot of knowledge about the specific task. As a consequence, algorithm developer's expectations and presumptions have an effect on the results. It would be profitable to have a similarity metric that is not dependent on the user.

The proposed method utilizes low bit rate audio coding, which retains the perceptually most relevant information of the signal. The similarity of two samples is estimated using compression based similarity measure. The proposed method does not require setting of any parameters and it is especially practical in applications where there is very little knowledge about the contents of the database beforehand.

The paper is organized as follows. Section 2 describes the overview of the system, Section 3 describes the signal representation, Section 4 presents the compression based similarity metric. Section 5 gives experimental results and comparisons to the other methods and finally Section 6 is for conclusions.

2. SYSTEM OVERVIEW

The overview of the system is illustrated in Fig. 1. First, perceptual audio coding (MP3, AAC etc) is applied to the original audio files. Second, the coded signals are compressed alone using some lossless compression method (gzip,bzip etc.). Third, the files are concatenated into a single file and compressed together using the same compression method. Finally, similarity is calculated by estimating the benefit achieved by compressing the files together.

When the similarity estimates are received, there are two application-dependent main possibilities how to return the results to the user. The first, referred as k-nearest neighbor query (k-NN) [5], is to sort the signals in order of the similarity and retrieve a fixed number of most similar samples to the user. A drawback is

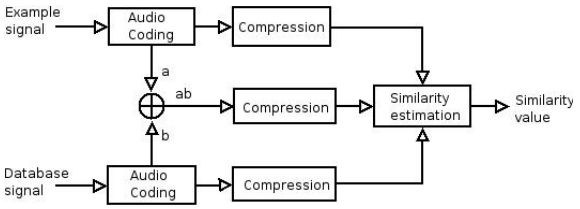

Figure 1: *Overview of the similarity estimation.*

that there is a possibility that some of the received samples are very different from the example, since a fixed number of samples is retrieved. Furthermore, the whole database have to be queried before the results can be presented.

The other possibility is to set a threshold, and retrieve all the samples that are closer than the threshold. This method is referred as ϵ-range query [5]. All samples inside the ϵ neighborhood of the query sample are retrieved. This way all the retrieved samples should be relatively similar to the example and similar samples may already be returned to the user during the query. The disadvantage of this method is that adjusting the threshold may not be straightforward and it might require user feedback or going through the whole database. In this study both methods are considered.

3. SIGNAL REPRESENTATION

The compression-based similarity measure requires a representation, where similar signals contain identical parts. A digital PCM signals are therefore too precise for this purpose. Perceptual audio coding provides a representation, where perceptually most important characteristics of a signal are retained and the signal is quantized so that identical codewords are present.

3.1. Perceptual audio codecs

Perceptual audio coding aims at representing an audio signal with a small amount of data while retaining the perceptual quality as close to the original as possible. Contrary to source coding, generic audio codecs remove the data which is perceptually irrelevant [6, pp. 41-42], thus they are lossy. They achieve compression by utilizing the properties of the human auditory system, especially the masking phenomenom. It refers to a situation where a separately audible sound becomes inaudible in the presence of a louder sound. The phenomenom is strong when the sounds occur simultaneously and are closely spaced in frequency.

General-purpose perceptual audio codecs are currently widely used in consumer electronics, for example in digital television, internet audio, and portable audio devices. The most commonly used codecs are developed in the standardization framework of Moving Picture Experts Group (MPEG). They include MPEG-1 Layer 3 (commonly known as MP3) and its successor Advanced Audio Coding (AAC). The perceptual codecs tested in this system include MP3 encoder LAME[1] and AAC encoder FAAC[2].

The basic idea of perceptual audio codecs is to quantize the input signal so that the quantization noise is inaudible. Since the masking phenomenon can be more easily modeled in time-frequency domain, codecs calculate a time-frequency representation using a filter bank or short-time frequency transforms. An auditory model approximates the masking effect, measures the audibility of the quantization noise, and controls the amount of bits required to represent the signal. The redundancy of the quantized codewords can be reduced by entropy coding.

[1] `http://lame.sourceforge.net/`

[2] `http://www.audiocoding.com/`

4. SIMILARITY MEASURE

To measure the similarity, we apply a measure developed by Bennett et al., which approximates the information distance between two sequences by compression [7]. The similarity measure has been previously used to a wide range of tasks: fetal heart rate tracings [8], classification of books by the author, optical character recognition, and building an evolutionary tree from mitochondrial genomes [9]. These studies show that the measure can be used in a wide range of application areas, and it does not need any specific knowledge about the task. Accuracy of such parameter-free algorithm is shown to be superior compared to traditional methods [10]. The distance used is referred as normalized compression distance, which is an estimate of normalized information distance.

4.1. Normalized compression distance

The minimum amount of information required to represent given string x is refered as Kolmogorov complexity. $K(x|y)$ is the conditional Kolmogorov complexity of string x relative to string y defined as the length of the shortest binary program to compute x if y is given as an auxiliary input. The minimum amount of information required to generate string x from string y and vice versa is refered as information distance (ID) [7]:

$$ID(x,y) = max\{K(x|y), K(y|x)\}. \tag{1}$$

This distance metric has two major drawbacks. First, it measures absolute distances meaning that two short random samples would have the same distance as two, almost similar, long samples. In order to have relative distance metric, the normalized information distance (NID) was proposed in [11]:

$$\mathrm{NID}(x,y) = \frac{max\{K(x|y), K(y|x)\}}{max\{K(x), K(y)\}}, \tag{2}$$

The other drawback is that this distance metric is based on the notions of Kolmogorov complexities, which are noncomputable. As a consequence, the approximation of the metric has to be used. The $K(x)$ and $K(y)$ are approximated here using $C(x)$ and $C(y)$, which are the sizes of compressed x and y respectively. The similarity between two signals is therefore approximated using a normalized compression distance (NCD) [9]:

$$\mathrm{NCD}(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}, \tag{3}$$

where $C(xy)$ is the compressed size of concatenated x and y. NCD is the measure of difference, thus larger values stand for more different signals. The value of NCD is between 0 and $1 + \epsilon$, since the compression techniques are not ideal.

This method can also be seen as parametric method, because the compression algorithm has to be chosen. However, the objective is to get the best approximation of the Kolmogorov complexity, therefore the algorithm that provides the best compression ratio should be chosen.

5. SIMULATION EXPERIMENTS

The performance of the proposed system was tested against other query by example methods. The methods were the Euclidean distance between the GMM densities [4], likelihood of GMMs [1], feature histogram based method [3], KL divergence of one-component GMMs [2], and Mahalanobis distance of feature means [2].

All these methods use the following preprocessing: first, signals are divided into 46 ms frames and second, several features are extracted from the frames. The feature set used here is the same as in [3] and [4]: Mel-frequency cepstral coefficients (three first coefficients), spectral spread, spectral flux, harmonic ratio, maximum autocorrelation lag, crest factor, noise likeness, crest factor, total energy, and variance of instantanous power. Before the processing, each feature is normalized to have zero mean and unity variance over the whole database.

Tested audio coding methods were MP3, AAC, and adaptive multi-rate (AMR). Bitrates between 8-64 kbits/s were tested and the best ones were chosen to be presented here. In AAC we used a version which does not apply frame wise Huffman coding to the signal, because this gave slightly better results than the original one. The method was also tested directly to wave files without any perceptual audio codec. Different lossless compression algorithms were also tested but the results were almost the same for all of them, the gzip is used in the simulations.

Simulations were carried out using an audio database which contains 1332 samples with 16 kHz sampling rate. The signals were manually annotated into 4 main classes and 17 sub classes. The classes and the number of samples in each class are listed in Table 1. Samples for the environmental class are taken from CASR recordings [12]. The subclasses correspond the classes in CASR (car, restaurant, road). The drum samples are acoustic drum sequences used by Paulus and Virtanen [13]. The rest of the music class are from RWC Music Database [14], acoustic class is from RWC Jazz Music Database, electroacoustic is from RWC Popular Music Database, and Symphony is from RWC Classical Music Database. Sing mainclass, which contains only monophonic singing, was taken from Vox database presented in [15]. The speech samples are from the CMU Arctic speech database [16].

All the samples in our database are 10 seconds long. The length of speech samples in Arctic database are 2-4 seconds, thus the samples from each speaker are combined to result in 10-second samples. Original samples in the other databases are longer than 10 seconds, thus random 10 second clips are cut from those.

Main class	Sub class
Environmental (231)	Inside car (151)
	In restaurant (42)
	Traffic (38)
Music (620)	Acoustic (264)
	Drums (56)
	Electroacoustic (249)
	Symphony (51)
Sing (165)	Humming (52)
	Singing (60)
	Whistling (53)
Speech (316)	Speaker1 (50)
	Speaker2 (47)
	Speaker3 (44)
	Speaker4 (40)
	Speaker5 (47)
	Speaker6 (38)
	Speaker7 (50)

Table 1: *Classes.*

5.1. Evaluation procedure

One signal at the time is drawn from the database to serve as a query signal. This query signal is compared against the other signals in database in order to find near similar samples. This procedure is repeated for 10 random signals from each class. Altogether $10(n-1) * number\_of\_classes$ comparisons are performed, where n is the total number of signals in the database. K-NN search and ϵ-range query were tested. If the example and retrieved signal are labelled in the same class, the database signal is seen as correctly retrieved from the database.

Averages of recall and precision rates of classes are used to present the results. Recall reveals the portion of similar signals retrieved from the database:

$$recall(class) = \frac{N_{ccs}}{n_{class}(n_{class}-1)}, \tag{4}$$

where n_{class} is the number of samples in the class, and N_{ccs} means the number of correctly retrieved samples from this class.

Precision gives the portion of correctly retrieved samples from all the retrieved signals:

$$precision(class) = \frac{N_{ccs}}{N_D}, \tag{5}$$

where N_D is the total number of samples retrieved from certain class when the example signal is from this class.

5.2. Results

The results from compression based method using different audio coding algorithms compared to other methods in k-NN search when $k = 20$ are presented in Table 2. The results of ϵ-range query with different values of ϵ are illustrated in Figure 2.

The proposed method outperforms the reference methods in ϵ-range query with large values of ϵ. This means it is the most accurate method when the aim is to retrieve all the similar samples from the database. In k-NN search using k=20, the results were also relatively good but slightly lower than with the best feature-based method.

There were only minor differences between different audio codecs, AAC resulting in the best average results. Using no audio codec at all gave very poor results. This was expected considering that compression algorithms require an identical strings to compress and in wave format already a very small change generates different codewords. Similar effect can be seen when using higher bitrates in audio codecs thus the lower bitrates gave the best results.

6. CONCLUSIONS

In this paper, a novel approach to query by example for audio signals was presented. First, perceptually important characteristics of a signal are retained by using a perceptual audio coder. Then

Coding method	Prec. main %	Prec. sub %
No codec	29.0	8.7
MP3 8 kbit/s	94.1	68.8
AMR 8 kbit/s	96.5	83.0
AAC 10 kbit/s	96.5	85.5
Mahalanobis distance	97.3	92.6
Likelihood of GMMs	94.0	86.8
Histogram method	85.6	75.4
Euclidean distance of GMMs	97.5	95.7
KL distance of GMMs	97.5	90.8

Table 2: *Precision values for main classes and sub classes for different audio coding methods, and feature based methods with k-NN search, when k=20. Gzip is used as a compressor.*

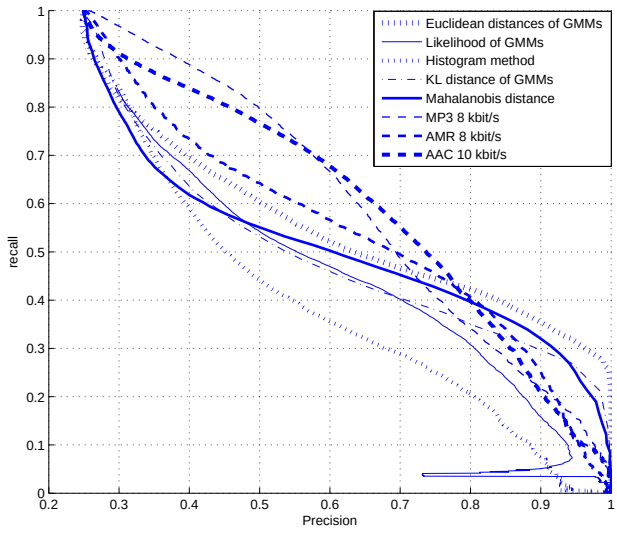

Figure 2: *ϵ-range results of different methods with different values of ϵ.*

coded audiofiles are compressed using standard lossless compression techniques and similarity is estimated from the compression ratios of individual files and combined files. The compression-based similarity metric does not require the setting of any parameters nor does it require any knowledge about the topic at hand.

The compression-based method was tested against the existing query by example methods. In ϵ-range query it outperformed the other methods at high recall rates and also in k-NN query it gave competitive results. This reveals that considering the simplicity of the proposed method, it is very practical for many applications. Especially ones, where there is very little knowledge about the contents of the database beforehand and thus, choosing the right features is impossible.

7. ACKNOWLEDGEMENTS

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence program 2006 - 2011).

8. REFERENCES

[1] E. Pampalk, *Computational Models of Music Similarity and their Applications in Music Information Retrieval*, Ph.D. thesis, Technische Universitat, Wien, 2006.

[2] M. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *Proc. 6th International Conference on Music Information Retrieval*, 2005.

[3] M. Helén and T. Lahti, "Query by example methods for audio signals," in *Proc. 7th IEEE Nordic Signal Processing Symposium*, Reykjavik, Iceland, June 2006, pp. 302–305.

[4] M. Helén and T. Virtanen, "Query by example methods of audio signals using Euclidean distance between Gaussian mixture models," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, USA, 2007.

[5] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi, "Approximate nearest neighbor searching in multimedia databases," in *Proc. 17th IEEE International Conference on Data Engineering*, Heidelberg, Germany, Apr. 2001.

[6] K. Brandenburg, "Perceptual coding of high quality digital audio," in *Applications of Digital Signal Processing to Audio and Acoustics*. 1998.

[7] C. H. Bennett, P. Gács, M. Li, P. M. B. Vitányi, and W. H. Zurek, "Information distance," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1407–1423, July 1998.

[8] C. Costa-Santos, J. Bernandes, P. M. B. Vitányi, and L. Antunes, "Clustering fetal heart rate tracings by compression," in *Proc. 19th IEEE International Symposium on Computer-Based Medical Systems*, Salt Lake City, Utah, USA, 2006.

[9] R. Cilibrasi and P. M. B. Vitányi, "Clustering by compression," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1523–1545, Apr. 2005.

[10] E. Keogh, S. Lonardi, and C. Ratanamahatana, "Towards parameter-free data mining," in *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, Aug. 2004.

[11] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[12] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florida, USA, May 2002.

[13] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," in *Proc. 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. 3rd International Conference on Music Information Retrieval*, Oct. 2002.

[15] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *Proc. 2003 Finnish Signal Processing Symposium (FINSIG'03)*, Tampere, Finland, May 2003, pp. 59–63.

[16] J. Kominek and A. Black, "The cmu arctic speech databases," in *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 223–224.