Content-Based Sound Retrieval for Web Application

Chunru Wan, Mingchun Liu, and Lipo Wang

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798
{ecrwan, P147508078, elpwang}@ntu.edu.sg

Abstract. It is both challenging and desirable to be able to retrieve sound files relevant to users' interests by searching the Internet. Unlike the traditional way of using keywords as input to search for web pages with relevant texts, query example can be used as input to search for similar sound files. In this paper, content-based technology has been applied to automatically retrieve sounds similar to the query-example. Features from time, frequency and coefficients domains are firstly extracted from each sound file. Next, Euclidean distances between the vectors of query and sample audios are measured. An ascending distance list is given as retrieval results. Experiments have been conducted on a sound database with 414 files from 16 classes. Further, we propose to classify the query audio into three classes, speech, music and other sound, with much fewer features and then search relevant files only in that subspace. This way, the retrieval performance could be further increased with the saving of computing time as well. Simulations show that our method leads to better results compared to the Soundfisher software in terms of both retrieval quality and completeness.

1 Introduction

Few search engines allow users to search the Internet with sounds as query inputs. However, users could benefit from the ability to direct access to these medias, which contain rich information but couldn't be precisely described by words.

Content-based retrieval thus has emerged from the wide spread of web applications. In the recent years, some researchers have applied this technology in audio retrieval. A general audio classification and retrieval system is built by Wold, et al, with a demo program online [1]. A nearest feature line method is proposed by Stan for the same kind of task [2]. In [3], a system for query a music database by humming is described along with a scheme for representing the melodic information in a song as relative pitch changes. Other techniques involved in audio retrieval such as structured representation of audio support for browsing, active search algorithm for quickly detect and locate known sounds, are described in [4] and [5] respectively.

In this paper, we retrieve similar sound files using Eclidean distance measurement between query and sample feature vectors. The rest of paper is organized

N. Zhong et al. (Eds.): WI 2001, LNAI 2198, pp. 389–393, 2001.

as follows. Feature extraction and normalization along with the database are discussed in section 2. The audio retrieval and its performance have been discussed in section 3. To further increase the performance and ease for browsing, the first hierarchy of audio directory can be automatically generated by k-nearest neighbor (k-NN) classifier, which is described in section 4. Finally, conclusions and discussions are included in section 5.

2 Feature Extraction

Feature extraction is the first step towards content-based retrieval. Here, we extracted features from time, frequency and coefficient domains and combine them to form a feature vector for each audio file in the database.

2.1 The Database

There are 414 sound files all together in the database, which forms 16 classes. All files are in 'au' format with sample rate of 8000Hz. The file length ranges from 0.25 second to less than ten seconds. A brief description of the database is given in Table 1.

Table 1. Structure of the database

Class name	File number	Class name	File number
1. Speech	53	Violin-pizzicato	40
Female	36	3. Sound	62
Male	17	Animal	9
2.Music	299	Bell	7
Trombone	13	Crowds	4
Cello	47	Laughter	7
Oboe	32	Machines	11
Percussion	102	Telephone	17
Tubular-bell	20	Water	7
Violin-bowed	45	Total	414

2.2 Feature Extraction

Since the audio files are short clips, segmentation is omitted. Each audio file is directly divided into frames of 256 samples, with 50% overlapping at the two adjacent frames. Silence reduction is done before the feature extraction. If the energy of an individual frame is below a predefined threshold, the whole frame is ignored.

After silence reduction, the audio frames are hamming-windowed and 48 features are extracted for each frame. Means and standard deviations of the frame-based features are computed as the final features for each file. The 48

Table 2. Structure of 48 extracted features

1. Time domain (9 features)	Mean and standard deviation of volume root mean square, zero-crossing ratio and frame energy. Volume dynamic ratio, silence ratio and total energy.
2.Frequency domain (26 features)	Mean and standard deviation of frequency centroid, bandwidth, four sub-band energy ratios, pitch, salience of pitch, first two formants, the amplitude at individual formants and spectrogram.
3.Coefficientdomain (13 features)	Mean of first 13 orders of MFCCs (Mel-Frequency Cepstral Coefficients).

features can be sorted into temporal, frequency and coefficients domains, as shown in Table 2.

After feature extraction, we normilize the feature values accross the whole database. Normalization can ensure that contributions of all audio feature elements are adequately represented. The magnitudes of the feature element values are more uniform after normalization and this will prevent particular feature from dominating the whole feature vector.

3 Audio Retrieval

The advent of world wide web has importance of information retrieval. It has increased the need for automated information retrieval for multimedia database in recent years. Here, we access and retrieve audio by its extracted feature vector. When a user inputs a query audio file and requests for finding relevant files to the query, both the query and each document in the database are represented as feature vectors. A measure of the similarity between the two vectors is computed, and then a list of files based on the similarity are fed back to the user for listening and browsing. The user may also refine the query to get more audio material relevant to his or her interest by relevant feedback.

To measure the performance of the contented based retrieval system, the precision and recall are used [6], as defined by

$$\text{Precision} = \frac{\text{relevant files retrieved}}{\text{total files retrieved}}, \tag{1}$$

$$\text{Recall} = \frac{\text{relevant files retrieved}}{\text{total files relevant}}. \tag{2}$$

The precision provides an indication of the quality of the answer set, while the recall considers the number of relevant documents retrieved. In an ideal situation, precision is always 1 at any recall point. In the experiment, we assume that the files in the same class are relevant; otherwise, they are non-relevant. Hence, the performance can be measured automatically without hearing the sound.

To test the performance of audio retrieval, the leave-one-out experiment is conducted. Each file is selected from the database as query and searched from the rest of files. Most often, users only browse the files ranked in the top list.

For this concern, top ten retrieved files for three queries from speech, music, and sound respectively are listed at the left column under each query in Table 3. The results are compared with soundfisher^{TM} software beta version 4, from Musclefish LLC [1] at the right column. Their precision and recall are also given in the table where only the top ten records are considered.

Table 3. Result of audio retrieval (compared with SoundFisher from Musclefish LLC)

Query	Female		Percussion		Laughter	
No.1	Female	Female	Percussion	Percussion	Laughter	Laughter
No.2	Female	Water	Percussion	Violinbowed	Laughter	Animal
No.3	Female	Female	Percussion	Telephone	Animal	Machines
No.4	Female	Male	Percussion	Oboe	Water	Percussion
No.5	Male	Male	Percussion	Telephone	Animal	Machines
No.6	Female	Percussion	Trombone	Cello	Animal	Laughter
No.7	Water	Female	Violinbowed	Percussion	Percussion	Crowd
No.8	Male	Female	Violinbowed	Animal	Percussion	Crowd
No.9	Male	Percussion	Violinbowed	Violinpizz	Laughter	Percussion
No.10	Male	Male	Percussion	Oboe	Laughter	Bell
Precision	0.5	0.4	0.6	0.2	0.4	0.2
Recall	0.14	0.11	0.06	0.02	0.67	0.33

4 Hierarchical Searching

For almost every existing text web search engine, there is a directory through which one can browse from a broader area to a more specific area. These directories are often built and maintained manually. We can automatically establish such a hierarchy for audio retrieval. For example, we can firstly classify audio files into three major classes: speech, music, and other sound. This is treated as the first level hierarchy.

A k-nearest neighbor (k-NN, k=2) classifier is used to classify the audio into the above three major classes. When all the 48 features are used, the accuracy is 96.6% of the testing set. After dropping one 'worst' feature, the accuracy increases. This may be caused by too high feature dimension for the given training patterns. We continue to drop 'worst' feature at a time, the classification accuracy varies up and down slightly along with the feature dimension. When the feature dimension reduced to 9, 100% accuracy can be achieved. If we further drop features, the performance will drop monotonically down to 83% with only one feature selected. Based on this observation, we chose the remained 9 features for classifying audio into the first hierarchy. After the classification, we then use more features to retrieve the files in a specific class in stead of the whole database. Thus, the precision and recall rates will be improved with less computation time.

5 Conclusions and Discussions

In this paper, content-based audio retrieval is conducted based on the features extracted from time, frequency and coefficient domains. The Euclidean distance measurement is used to search sound files relevant to the query example. Further, an audio hierarchy is built by k-NN classifier using a small set of features from the whole feature vector. Thus query audio can be firstly classified into speech, music, or sound, and then retrieve files only in the subspace. Consequently, precision and recall rates could be improved with the saving of computing time as well. Simulations show that our method leads to better results compared to the Soudfisher software in terms of both retrieval quality and completeness.

For a real system, in case of long files, we can segment them into smaller segments and do classification individually. The recognized classes can be taken as terms, which is the key element in text retrieval. With obtained terms, some text retrieval strategies can be applied to audio retrieval and searching by using both keywords and query example. How to extract key clips for long audio file and deal with database growing dynamically are other challenging fields for audio browsing. Future content-based audio retrieval system must also provide access to conceptual semantics not just low level features.

References

1. Wold, E., Blum, T., Keislar, D., et. al.: Content-based classification, search, and retrieval of audio. IEEE Multimedia, (1996) 27–36,
2. Li, S. Z.: Content-Based Classification and Retrieval of Audio Using the Nearest Feature Line Method. IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 5, (2000) 619–625
3. Ghias, A., Logan, J., Chamberlin, D., Smith, B.C.: Query by humming: musical information retrieval in an audio database. Proceedings of the third ACM international conference on Multimedia, (1995) 231–236
4. Melih. K., Gonzalez, R.: Audio retrieval using perceptually based structures. Proceedings of IEEE International Conference on Multimedia Computing and Systems, (1998) 338–347
5. Smith, G., Murase, H., Kashino, K.: Quick audio retrieval using active search. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol 6, (1998) 3777–3780
6. Grossman, D.A., Frieder, O.: Information Retrieval: Algorithms and Heuristics. Kluwer Academic Publishers (1998)