Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York, Oct. 17-20, 1999

MULTIFEATURE AUDIO SEGMENTATION FOR BROWSING AND ANNOTATION

George Tzanetakis

Computer Science Department
Princeton University
35 Olden Street, Princeton NJ 08544,USA
gtzan@cs.princeton.edu

Perry Cook

Computer Science and Music Deptartment
Princeton University
35 Olden Street, Princeton NJ 08544,USA
prc@cs.princeton.edu

ABSTRACT

Indexing and content-based retrieval are necessary to handle the large amounts of audio and multimedia data that is becoming available on the web and elsewhere. Since manual indexing using existing audio editors is extremely time consuming a number of automatic content analysis systems have been proposed. Most of these systems rely on speech recognition techniques to create text indices. On the other hand, very few systems have been proposed for automatic indexing of music and general audio. Typically these systems rely on classification and similarity-retrieval techniques and work in restricted audio domains.

A somewhat different, more general approach for fast indexing of arbitrary audio data is the use of segmentation based on multiple temporal features combined with automatic or semi-automatic annotation. In this paper, a general methodology for audio segmentation is proposed. A number of experiments were performed to evaluate the proposed methodology and compare different segmentation schemes. Finally, a prototype audio browsing and annotation tool based on segmentation combined with existing classification techniques was implemented.

1. INTRODUCTION

The increasing amounts of multimedia data pose new challenges to information retrieval (IR). Traditional IR, familiar from the popular Web search engines, offers the ability to quickly locate and browse large amounts of computer-readable text using the familiar search and "ranked-by-similarity" interface. Unfortunately, for multimedia there are no widely used similar techniques.

There are two main approaches for multimedia information retrieval. The first is to generate textual indices automatically, semi-automatically or manually and then use traditional IR. The other approach is to use content-based retrieval were the query is non-textual and a similarity measure is used for searching and retrieval.

An example of the first approach is the Informedia project [1] where a combination of speech-recognition, image analysis and keyword searching techniques is used to index a terrabyte video archive.

A retrieval-by-similarity system for isolated sounds has been developed at Muscle Fish LLC [2]. Users can search for and retrieve sounds by perceptual and acoustical features, can specify classes based on these features and can ask the engine to retrieve similar or dissimilar sounds.

In this paper we focus on audio data and especially music. Recently, a number of techniques for automatic analysis of audio information have been proposed [3]. These approaches work reasonably well for restricted classes of audio and are based on pattern recognition techniques for classification. Unlike these methods, we directly segment audio into regions based on temporal changes without trying to classify the content.

The remainder of this paper is comprised of 5 sections. Section 2 describes a general segmentation methodology based on multiple temporal features. In section 3, the experiments performed to evaluate the methodology are described. In section 4, a prototype audio browser that combines segmentation and classification for fast browsing and annotation is described. Future work is discussed in section 5.

2. SEGMENTATION

2.1. Motivation

One of the first chapters of most textbooks in image processing or computer vision is devoted to edge detection and object segmentation. This is because it is much easier to build classification and analysis algorithms using as input segmented objects rather than raw image data. In video analysis, shots, pans and generally temporal segments are detected and then analyzed for content. Similarly temporal segmentation can be used for audio and especially music analysis.

Auditory scene analysis is the process by which the human auditory system builds mental descriptions of complex auditory environments by analyzing mixtures of sounds [4]. From an ecological viewpoint, we try to associate events with sounds in order to understand our environment. The characteristics of sound sources tend to vary smoothly in time. Therefore abrupt changes usually indicate a new sound event. The decisions for sequential and simultaneous integration of sound are based on multiple cues. Although our method does not attempt to model the human auditory system, it does use significant changes of multiple features as segmentation boundaries. The experiments indicate that the features selected contain enough information to be useful for automatic segmentation.

Temporal segmentation is a more primitive process than classification since it does not try to interpret the data. Therefore, it can be more easily modeled using mathematical techniques. Being more simple it can work with arbitrary audio and does not pose specific constraints on its input like single speaker or isolated tones. It has been argued in [5] that music analysis systems should be built for and tested on real music and be based on perceptual properties rather than music theory and note-level transcriptions.

Annotation of simple cases like musical instruments or music vs speech can be performed automatically using current classification systems. Based on these techniques, a completely automatic annotation system for audio could be envisioned. Although not impossible in theory, there are two problems with such an approach. The first is that current systems are not perfect and, therefore, annotation errors are inevitable. This problem has to do with the current state of the art, so it is possible that in the future it will be solved. There is a second problem, however, that is more subtle and not so easy to address. Audio, and especially music, is heard and described differently by each listener. There are, however, attributes of audio that most listeners will agree upon, like the general structure of the piece, the style, etc. Ideally a system for annotation should automatically extract as much information as it can and then let the user edit and expand it.

This leads to a semi-automatic approach that combines both manual and fully automatic annotation into a flexible, practical user interface for audio manipulation. Automatic segmentation is an important part of such a system. For example, the user can automatically segment audio into regions then run automatic classification algorithms that suggest annotations for each region. Then the annotations can be edited and/or expanded by the user. This way, significant amounts of user time are saved without loosing the flexibility of subjective annotation. The audio browser described in this paper, is an initial prototype of such a semi-automatic system.

2.2. Related work

Hidden Markov Models were used in [6] for segmentation and analysis of recorded meetings by speaker. The breakup into segments is based on classification and assumes that a trained model for each speaker is available. The trajectory of the fundamental frequency is used in [7] for segmenting voice into phonemes or notes. In contrast, our method does not need any training and can use multiple features for segmenting audio and especially music.

2.3. General Methodology

A general methodology for temporal segmentation based on multiple features is described. Using this methodology different segmentation schemes can be designed depending on the choice of parameters, features and heuristics. In order to evaluate the proposed methodology we implemented and tested some of these schemes. The main goal was to demonstrate empirically the validity of the approach and provide a general framework for constructing segmentation algorithms rather than finding an optimal scheme.

The method can be broken into four stages. By abstracting the basic steps it describes a family of possible segmentation schemes. Specific details for particular schemes we used can be found in the following subsections.

1. A time series of feature vectors V_t is calculated by iterating over the sound file. Each feature vector can be thought of as a short description of the corresponding frame of sound.
2. A distance signal $\Delta_t = ||V_t - V_{t-1}||$ is calculated between successive frames of sound. In our implementation we use a Mahalonobis distance. It is defined by

$$D(x, y) = (x - y)^T \Sigma^{-1} (x - y) \tag{1}$$

   where Σ is the feature covariance matrix calculated from the whole sound file. This distance rotates and scales the feature space so the contribution of each feature is equal. Other distance metrics, possibly using relative feature weighting, can also be used.
3. The derivative $\frac{d\Delta_t}{dt}$ of the distance signal is taken. The derivative of the distance will be low for slowly changing textures and high during sudden transitions. Therefore the peaks roughly correspond to texture changes.
4. Peaks are picked using simple heuristics and used to create the segmentation of the signal into time regions. As a heuristic example, a minimum duration between successive peaks can be set to avoid small regions.

2.4. Features

It is typical for audio and speech analysis algorithms to be based on features computed on a frame basis. This is necessary to reduce the amount of data to be processed as well as the variability. These features can be thought of as a short term description of the sound for that particular moment in time. For example MFCC [8] (Mel-Frequency Cepstral Coefficients) characterize the vocal tract resonances and are commonly used in speech recognition. Since our methodology is based on frame-based features it is easy to use existing front-ends of other applications and extend them with segmentation.

2.5. A specific scheme

Following this approach, a segmentation scheme was implemented using a similar feature front-end as the music/speech discriminator described in [9, 7]. This scheme was used for the experiments. Basic features are calculated every 20 msec. The actual features used are the means and variances of these features in a 1 sec window. The five basic features (resulting in ten actual features) are:

Spectral Centroid is the balancing point of the spectrum. It can be calculated using

$$C = \frac{\sum_i i A_i}{\sum_i A_i} \tag{2}$$

where A_i is the the amplitude of frequency bin i of the spectrum.

Spectral Rolloff The 95 percentile of the power spectral distribution. This is a measure of the "skewness" of the spectral shape.

Spectral Flux is the 2-norm of the difference between the magnitude of the Short Time Fourier Transform (STFT) spectrum evaluated at two successive sound frames. The STFT is normalized in energy.

ZeroCrossings is the number of time-domain zero-crossings. It is a correlate of the spectral centroid.

RMS is a measure of the loudness of the frame. This feature is unique to segmentation since changes in loudness are important cues for new sound events. In contrast, classification algorithms must be loudness invariant.

Similar features are also used in [2] for similarity-retrieval of isolated sounds. More detailed descriptions of the features can be found in [9, 7, 2].

The peak picking heuristic is parameterized by the desired number of peaks. This was a necessary property for the experiments in this paper comparing human and automatic segmentation. The heuristic is described by the following algorithm:

1. The peak with the maximum amplitude is picked.
2. A region around and including the peak is zeroed (helps to avoid counting the same peak twice). The size of the region is proportional to the size of sound-file divided by the number of desired regions (20% of the average region size)
3. Step 1 is repeated until the desired number of peaks is reached.

3. EXPERIMENTS-EVALUATION

A pilot study was conducted to explore what humans do when asked to segment audio and to compare those results with the automatic segmentation method. Evaluating segmentation performance is difficult because there is no mathematical criterion of how well the algorithm performs.

In order for any automatic segmentation to be useful we must first make sure that humans are consistent when segmenting audio. Then the results of the automatic method are compared with the human results.

The data used consists of 10 sound files about 1 minute long. A variety of styles and textures are represented. Nine subjects were asked to segment each sound file using standard audio editing tools in 3 ways. The first way, which we call free, is breaking up the file into any number of segments. The second and third way constrain the users to a specific budget of total segments 4 ± -1 and 8 ± 2.

The results are shown in Tables 1,2 and 3. The segments that more than 4 of the 9 subjects agreed upon were used for comparison. The AG column shows the number of these salient segments compared to the total number of all segments marked by any subject. It is a measure of consistency between the subjects. For comparing the automatic method a segment boundary was considered to be the same if it was within 0.5 sec of the average human boundary. This was based on the deviation of segment boundaries between subjects. FB (fixed-budget) refers to automatic segmentation by requesting the same number of segments as the salient human segments. BE (best effort) refers to the best automatic segmentation achieved by incrementally increasing the number of regions up to a maximum of 16. MX is the number of segments necessary to achieve the best effort segmentation.

The results show that humans are consistent when segmenting audio (more than half of the segments are common for most of the subjects). In addition, they show that human segmentation can be approximated by automatic algorithms. The biggest problem seems to be the perceptual weighting of a texture change. For example, many errors involved soft speech entrances that were marked by all the subjects although they were not significant as changes in feature space. The automatic segmentation results are usually perceptually justified and a superset of the human segmentated regions.

More detailed results in addition to the segmented sound files are available on the Web:
http://www.cs.princeton.edu/ gtzan/marsyas/results.html

4. AUDIO BROWSER

The typical "tape-recorder" paradigm for audio user interfaces is time-consuming and inflexible. For example, 2 hours was the average time required by the subjects of the experiment to manually segment and annotate 10 minutes of audio using standard sound editing tools. The main problem is that typical sound tools treat

	Human Agreement		Automatic				
			Fixed Budget		Best Effort		
	AG	**%**	**FB**	**%**	**BE**	**MX**	**%**
Classic1	8/14	57	7/8	87	7/8	8	87
Classic2	7/11	63	5/7	71	6/7	14	85
Classic3	5/13	38	2/5	40	3/5	16	60
Jazz1	3/14	21	2/3	66	3/3	16	100
Jazz2	5/8	62	3/5	60	5/5	10	100
JazzRock	6/9	66	0/6	0	5/6	12	83
Pop1	5/6	83	4/5	80	5/5	10	100
Pop2	5/10	50	4/5	80	4/5	5	80
Radio1	4/10	40	3/4	75	4/4	10	100
Radio2	8/11	72	5/8	62	6/8	11	75
Total	**56/106**	**55**	**35/56**	**62**	**48/56**	**11**	**87**

Table 1: Free segmentation

	Human Agreement		Automatic				
			Fixed Budget		Best Effort		
	AG	**%**	**FB**	**%**	**BE**	**MX**	**%**
Classic1	4/6	66	0/4	0	4/4	10	100
Classic2	4/7	57	3/4	75	3/4	4	75
Classic3	4/7	57	2/4	50	2/4	4	50
Jazz1	3/11	27	2/3	66	3/3	16	100
Jazz2	5/6	83	3/5	60	5/5	10	100
JazzRock	5/7	71	1/5	20	4/5	12	80
Pop1	4/7	57	2/4	50	4/4	10	100
Pop2	5/7	71	4/5	80	4/5	5	80
Radio1	4/6	66	3/4	75	4/4	10	100
Radio2	5/7	71	2/5	40	4/5	10	80
Total	**43/71**	**62**	**22/43**	**51**	**37/43**	**9**	**86**

Table 2: 4 ± 1 segmentation

	Human Agreement		Automatic				
			Fixed Budget		Best Effort		
	AG	**%**	**FB**	**%**	**BE**	**MX**	**%**
Classic1	8/14	57	7/8	87	7/8	8	87
Classic2	7/10	70	5/7	71	6/7	14	85
Classic3	9/11	81	6/9	66	6/9	9	66
Jazz1	4/15	26	3/4	75	3/4	4	75
Jazz2	5/11	45	3/5	60	5/5	10	100
JazzRock	7/9	77	3/7	42	5/7	12	71
Pop1	6/12	50	5/6	83	6/6	10	100
Pop2	8/13	61	4/8	50	5/8	16	62
Radio1	7/10	70	3/7	42	4/7	10	57
Radio2	9/11	81	5/9	55	6/9	11	66
Total	**70/116**	**61**	**44/77**	**63**	**53/70**	**10**	**77**

Table 3: 8 ± 2 segmentation

audio as a monolithic block of samples. Speech Skimmer [10] is an example of pushing audio interaction beyond the tape-recorder metaphor. The user can audition spoken documents at several times real-time, using time compression techniques and segmentation based on pitch.

Our prototype audio browser looks like a typical wave-form editor following the "tape-recorder" paradigm. However, in addition to the usual play, rewind, forward buttons, the user can segment the audio into regions either manually or using the described automatic segmentation methods. Forward and backward region buttons are provided for easy browsing. Each region can be annotated with text. Skipping and annotating using regions is much faster than manual annotation, in the same way that finding a song on a CD is much faster than finding it on a tape. Moreover using segmented audio the user can not only locate songs but also smaller units like a solo or a chorus.

In addition to demonstrating the possibilities of a better user interaction with audio, the browser aided in conducting the experiments. Being able to change the parameters,features and heuristics and hear the results under a unified GUI accelerated the cycle of implementation, experimentation and evaluation.

5. FUTURE WORK

5.1. Segmentation

In our implementation the importance of each feature is equal. Most likely this is not the case with humans and is not optimal. Therefore, relative weighting of the features needs to be explored. Optimization of the parameters and heuristics needs to be done. Moreover, different front-ends to segmentation can be used. An interesting case is the use of the MPEG audio analysis filter bank as a basis for calculating features. That way MPEG compressed audio data could be used directly for segmentation or classification. Another interesting direction is the use of a more perceptually motivated front-end for feature calculation like a cochlear model [11].

Based on the fact that on average the subjects needed more than 2 hours for segmenting 10 minutes of audio using standard audio editing tools we plan to compare how much this process can be accelerated using our automatic segmentation-based audio browser.

Because of the difficulty of evaluating and comparing different segmentation schemes standard corpora of audio examples need to be designed. Furthermore, we believe that the segmentation results combined with the calculated feature vectors should be used as an intermediate representation for further higher level analysis like classification or similarity-retrieval.

5.2. Applications

Segmentation can be used as a building block for automatic or semi-automatic music analysis. As a simple example, the structure of cyclic pop songs can be revealed using our segmentation scheme. A combination of segmentation with beat tracking methods [12] can offer significant information for music style identification and music analysis.

Another interesting application is audio thumb nailing. For each region, a characteristic segment has to be selected. These segments can be then used to create a shorter summary version of the original sound file.

6. CONCLUSIONS

We describe a general methodology for temporal audio segmentation based on multiple features. A prototype audio browser based on segmentation and classification was implemented and used to evaluate different segmentation schemes. A number of experiments were performed to compare human and automatic segmentation. The results indicate that human subjects are consistent when segmenting audio and it is possible to automate this process. We believe that our work shows the potential of automatic segmentation for audio analysis and hope it will stimulate more research in this direction.

7. REFERENCES

[1] A. Hauptmann and M. Witbrock, "Informedia: News-on-demand multimedia information acquisition and retrieval," in *Intelligent Multimedia Information Retrieval*, chapter 10, pp. 215–240. MIT Press, Cambridge, Mass., 1997, http://www.cs.cmu.edu/afs/cs/user/alex/www/.

[2] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 2, pp. 27–36, 1996.

[3] J. Foote, "An overview of audio information retrieval," *ACM Multimedia Systems*, vol. 7, pp. 2–10, 1999.

[4] A. Bregman, *Auditory Scene Analysis*, MIT Press, 1990.

[5] K. Martin, E. Scheirer, and B. Vercoe, "Musical content analysis through models of audition," in *Proc.ACM Multimedia Workshop on Content-Based Processing of Music*, Bristol, UK, 1998.

[6] J. Boreczky and L. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," *Proc. Int.Conf on Acoustics,Speech and Signal Processing Vol.6*, pp. 3741–3744, 1998.

[7] S. Rossignol, X. Rodet, et al., "Features extraction and temporal segmentation of acoustic signals," *Proc. ICMC 98*, pp. 199–202, 1998.

[8] M. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," in *Proc. 1996 ICASSP*, 1980, pp. 880–883.

[9] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," *IEEE Transactions on Acoustics, Speech and Signal Processing (ICASSP'97)*, pp. 1331–1334, 1997.

[10] B. Arons, "Speechskimmer: A system for interactively skimming recorded speech," *ACM Transactions Computer Human Interaction*, vol. 4, pp. 3–38, 1997, http://www.media.mit.edu/people/barons/papers/ToCHI97.ps.

[11] M. Slaney and R. Lyon, "On the importance of time-a temporal representation of sound," in *Visual Representations of Speech Signals*, M Cooke, B Beet, and M Crawford, Eds., pp. 95–116. John Wiley & Sons Ltd, 1993.

[12] E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J.Acoust.Soc.Am*, vol. 103, no. 1, pp. 588,601, Jan 1998.