A METHOD FOR DIRECT AUDIO SEARCH WITH APPLICATIONS TO INDEXING AND RETRIEVAL

S.E. Johnson & P.C. Woodland

Cambridge University Engineering Department,Trumpington Street, Cambridge, CB2 1PZ, UK.
Email: {sej28,pcw}@eng.cam.ac.uk

ABSTRACT

A technique for searching audio data to find an exact match for a given piece of *cue-audio* is described. The method uses a cepstral parameterisation of the audio and a covariance-based distance metric to quickly locate direct repeats. Results on data from ABC news broadcasts show that the method can successfully locate matches several hundred times faster than real-time and requires less than a second of cue-audio. By applying the match recursively to the data, repeated sections of audio, which nearly always correspond to *non-news* items such as commercials and theme-music, can be identified. Experiments show that the application of the technique can also lead to improved information retrieval using automatically transcribed broadcast data.

1. INTRODUCTION

With the ever-increasing size and availability of audio databases, it is necessary to have accurate and efficient methods of classifying, searching for and retrieving information from audio sources.

For some problems the most appropriate representation of the audio uses automatically derived text transcriptions. This approach to audio search has been investigated in the NIST Spoken Document Retrieval (SDR) evaluations[4, 5], by applying text-based retrieval methods to documents produced by a large vocabulary speech recognition system and it has been shown that such techniques can work well. However the transcription systems used (see e.g. [6]) typically run rather slower than real-time and have a word error rate of over 20%. Furthermore all information about music, laughter, background events, speaker etc. that are present in the audio are lost during the transcription process. Therefore for tasks in which precise details of the original audio is necessary, the transcription based methods are inappropriate.

Recent work in content-based audio retrieval addresses this issue by allowing sections of audio which are *similar* (in some sense) to a piece of *cue-audio* to be retrieved from a database. This is usually done by specifying some features to represent the audio and computing an average feature vector over a sound clip. A standard distance measure is then used to compare the clips to the cue-audio and a list of clips ranked by similarity is presented to the user.

While the ability to recognise similar sounds is useful for audio retrieval, classification and segmentation, there are several applications where an exact match is required. Naturally, similar techniques can be used, but the problem switches to a *filtering*-type task, namely a binary choice of whether the cue-audio is present or absent within a particular audio sample.

This paper addresses the problem of finding exact matches in audio segments in an efficient and accurate manner and the method can be applied to a wide range of applications. The techniques described can be particularly beneficial when dealing with broadcast material, since pre-recorded portions of audio which represent certain events are often re-broadcast several times. Therefore finding the location of such re-broadcast material can often reveal some broadcast structure. A simple example is the location of the end-point of a broadcast using a a single piece of cue-audio from the broadcaster. A more complex example is to automatically detect repeated events in a database of broadcasts which usually correspond to *non-news* items such as commercials or jingles. The automatic detection and elimination of non-news events can be usefully applied in an SDR system by indicating potential story or topic boundaries, increasing precision during retrieval and reducing recognition time.

This paper describes the search method in section 2 including the audio representation and distance metric used. The data used for experiments is described briefly in section 3 and the results for searching for a single piece of cue-audio indicating the end of a news show are given in section 4. Results for detecting non-news information by applying this technique recursively are given in section 5 and finally conclusions are offered in section 6.

2. DIRECT AUDIO SEARCH

2.1. Background

To be able to search audio accurately and quickly, the data must first be sampled and a set of features which capture the characteristics of the audio must be defined. Whilst it is possible in theory just to use the amplitude of the audio in the time-domain directly after sampling, more complicated features such as FFT-based coefficients or zero-crossing rate require several samples to be grouped together into a single *frame* (typically computed over 25ms with a 10ms frame shift). A single feature vector is then calculated for each frame within the *segment* of audio being considered.

Several methods of representing the entire segment of audio have been investigated. These range from calculating the mean, variance and small-lag autocorrelation of the frame-level feature vectors (e.g. in [9]), through using a histogram to represent the overall distribution of each feature (e.g. in [8]), to using the relative outputs from a discriminatively-trained vector quantiser (in [3]). The ability of the final representation to capture the specific properties of the piece of audio therefore depends on both the choice of initial features, and the method of representing these features over the entire segment.

Finally, a distance metric is required to calculate the similarity between segments. Metrics which have been used include the covariance-weighted Euclidean distance or the Cosine distance between the representative feature vectors. Audio retrieval systems present a ranked list of similar segments, whereas exact-search methods must compare the distance to a threshold (to compensate for asynchronous sampling or framing) to decide if the cue-audio is present.

2.2. Search Method

The audio representation used in the experiments reported here uses the mel-frequency PLP cepstral coefficients[10] that we also use in our large vocabulary speech recognition system. For applications in which the audio is also transcribed this data is available at no additional computational cost for use in direct audio search. We have investigated representing each frame of data with either just the "static" cepstra or also augmenting the representation with the first and second order times derivatives to capture dynamic features in the audio stream.

We have previously shown that the *arithmetic harmonic sphericity* (AHS) distance[1] between the covariance matrix of two segments (sequences of frames) of broadcast audio data can be successfully used for the speaker-identification task[7]. Capturing generic similarity between pieces of audio is important for speaker identification and similar techniques can therefore often be used for both speaker identification and content-based retrieval (as noted in [2]). Therefore the covariance-based comparison method using the AHS distance was used for the audio search task. The method compares the covariance matrix of the cue audio with that of segments of data of the same size in the audio data to be searched.

First the covariance matrix[1] for a segment is calculated from its D-dimensional feature vectors. The AHS distance between two segments is then defined as:

$$d(X,Y) = \log[tr(\Sigma_y \Sigma_x^{-1}) \cdot tr(\Sigma_x \Sigma_y^{-1})] - 2\log(D)$$

where tr represents the trace of the matrix. We have also evaluated a variant of the method in which the Euclidean distance between the elements of a diagonal covariance matrix is used.

Using the AHS measure, the distance between identical segments is zero. However, since there is no guarantee that the cue-audio event occurs synchronously with the frames in the audio being searched, a small threshold is defined and distances less than the threshold used to indicate a match.

When attempting to find a small piece of cue-audio in a long portion of target-audio, the latter is divided into *windows* of the same length as the former. Since the covariance-based representation attempts to capture the generic properties of the audio, it is no longer necessary to shift the search window by just a single frame and therefore the search speed can be considerably increased by introducing a *shift* between target-audio windows. The size of this shift depends on the length of cue-audio and the required trade-off between time resolution in locating the audio, accuracy of detection and searching speed. The experiments reported below show that when the cue-audio is longer than one second, the shift period can be half the window length with no drop in accuracy, confirming the ability of the covariance-based method to capture the generic properties of the audio.

[1]For these experiments we use the covariance matrix, but the mean information can easily be included by defining Σ as the correlation matrix.

3. DATA USED FOR EVALUATION

The experiments reported in this paper use the data recorded from broadcasts of the ABC World News Tonight which were included in the 1998 TREC-7 spoken document retrieval evaluation[4]. This set consists of 19 broadcasts from January 1998, each recorded for 31 minutes, although the average length of the desired show is only 27.5 minutes long, due to inaccurate start and end-pointing of the programmes.

Of the 8.7 hours of data within the desired shows, 6.2 hours has been labelled as `STORY` (a topic being discussed in the news) or `FILLER` (a general overview or topic shift) by human assessors. These groups were considered to contain potentially useful *news* information. The remaining 2.5 hours, which consists of introductions, music and commercials, were considered to be *non-news* bearing.

4. EXPT. 1 – END POINT DETECTION

The initial evaluation of the direct audio search procedure was carried out in the context of trying to automatically detect the endpoint of a broadcast news episode.

The majority of ABC episodes within the TREC-7 collection ended with the 6.63 second long pre-recorded announcement "This has been a presentation of ABC News. More Americans get their news from ABC News than any other source". The search method described in section 2, was conducted over the last 5 minutes of the audio for each 31 minute show using a sliding window with a 0.5s inter-window shift. The time for the best (i.e. smallest distance) match along with the corresponding distance was recorded for all 19 ABC episodes.

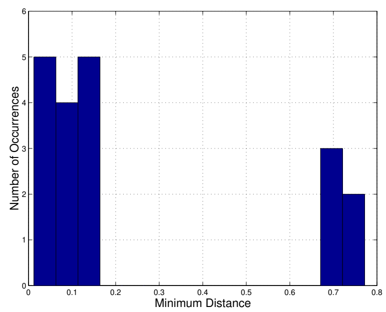

Figure 1: Distribution of minimum AHS distance for audio search on ABC episodes.

A histogram of the minimum distances when using the arithmetic harmonic sphericity distance is given in Figure 1. The distribution is clearly bi-modal with a large separation between the 14 shows where the cue-audio was present, and the 5 shows where it was absent.[2]

To further investigate the properties of the technique, the experiment was repeated using a simple Euclidean distance with the diagonal covariance matrix. The *separability* ratio, defined as the smallest minimum distance when the cue-audio was absent to the largest minimum distance when the cue-audio was present, the RMS error between the predicted start time and actual start time (when the cue-audio was present) and the speed of searching[3] are given in Table 1. Results when using no derivatives in the feature vector are also given for comparison.

Features Used	Dist.	Cov.	Speed (xRT)	RMS Error(s)	Sep. Ratio
+deriv	AHS	Full	0.0256	0.189	4.53
+deriv	Euc.	Diag.	0.0083	0.148	1.50
static	AHS	Full	0.0053	0.173	3.01
static	Euc.	Diag.	0.0030	0.148	1.50

Table 1: Speed, error and separability ratio for a window shift of 0.5s

These results show that all of the variants give a RMS error of below 0.2 seconds. Removing the derivatives from the feature vector leads to a 5-fold increase in speed, but the separability of the true and false matches is reduced from 4.53 to 3.01. The Euclidean-based runs give a low RMS error, but the separability ratio is rather small.

4.1. Speeding up the Search

To speed up the search, a threshold can be used to define when a match has been found (rather than continuing to search for the minimum score). To enable this threshold to be set robustly, the separability between the shows where the cue-audio is absent and those where it is present must be high. An experiment was conducted to see the effect on separability of speeding up the search by increasing the shift between the target-audio sliding windows. The results are given in Table 2.

Window Shift	AHS +deriv	AHS static	Euclid static
0.5s	4.53	3.01	1.50
1.0s	4.06	2.45	1.23
2.0s	2.36	1.73	1.01
3.0s	2.18	1.56	–
4.0s	2.27	1.60	–
5.0s	1.87	1.17	–

Table 2: Effect on separability of changing the window shift

[2]Two of these shows ended in silence due to a fault in recording, whilst the other three were broadcast on Sundays where the cue-audio was not used.

[3]The time does not include calculating the initial feature vectors. Measured on a 550MHz Pentium III.

These results show that the 6.6 second piece of cue-audio can be correctly identified every time it occurs, even when the window shift is increased to 5 seconds. The closest match when the cue-audio does not occur still gives a distance of almost twice the distance for a correct match and thus allows a more robust threshold to be set to indicate when a genuine match has occurred.

When using this technique, 1669 seconds of unwanted audio data occurring after the end-point of the show has been detected can be removed from for the 14 shows which contain the cue-audio in 7.8 CPU seconds.[4]

4.2. Reducing the Size of Cue-Audio

In many situations the cue-audio may be less than 6.6 seconds long. To investigate the robustness of the method when the size of cue-audio was reduced, an experiment was carried out with the results given in Table 3.

Size (s)	Shift (s)	Misses	Speed (xRT)	Sep. Ratio
6.0	0.5	0	0.0050	3.45
4.0	0.5	0	0.0040	2.62
2.0	0.5	0	0.0030	2.26
1.0	0.5	0	0.0026	1.53
0.5	0.25	3	0.0023	1.02
0.5	0.1	0	0.0041	2.18
0.3	0.1	0	0.0037	1.51
0.2	0.1	1	0.0034	1.07
0.2	0.05	1	0.0050	1.06

Table 3: Effect on accuracy, speed and separability of reducing the size of the cue-audio

These results confirm the power of the technique and show that a 1s piece of cue audio can still be reliably detected found with a separability of 1.5 for a window shift of 0.5s, whilst a similar separability can be achieved for a 0.3s piece of cue-audio if the window shift is reduced to 0.1s. The small amount of audio needed for this direct match is rather less data than would be needed for reliable location when only the *transcriptions* of the data are available. This provides another reason for using the direct audio search, even when completely accurate transcriptions are available.

5. EXPT. 2 – FINDING NON-NEWS EVENTS

Since pre-recorded commercials and jingles are often re-broadcast several times, the audio corresponding to non-news items such as commercials will often appear more than once in a given broadcast news database. Although news stories may be repeated, generally similar news is *re-read*, and hence although the words may be the same, the prosody, emphasis, timing, speaker, background condition and channel condition may all be different, leading to different audio characteristics. The presence of identical audio within a broadcast news database may therefore offer a good indication of the occurrence of non-news events which are of no interest to the average database browser.

[4]When using static coefficients only, with a window shift of 4 seconds and a threshold of 0.4

The ABC TREC-7 PLP data was analysed into fixed length overlapped *windows* and all the non-overlapping windows were compared to each other using the method described in section 2. Those windows which gave a distance less than a small threshold to another non-overlapping window were labelled as `non-news` events.

The results for three different window lengths and shifts are given in Table 4. Also provided for comparison are the results from recursively searching the segments generated automatically during the first stage of our transcription process[6]. There are fewer of these segments, since they do not overlap and some music and silence has already been eliminated during segmentation. Since the search is $O(N^2)$ using segments is beneficial if the segmentation has already been conducted and speed is critical, but otherwise should not be used.

Method	CPUs	non-news	news
window, 5@1s	3514	5459s=60.8%	110s=0.5%
window, 10@2s	967	5480s=61.0%	114s=0.5%
window, 20@4s	284	4598s=51.2%	122s=0.5%
Automatic segs*	38	3032s=33.8%	29s=0.1%

* After segmentation removed 18.3% non- and 1.2% news

Table 4: Amount of non-news and news information rejected on the ABC TREC-7 SDR data by recursively searching for repeated audio

Of the 110 seconds worth of "news" rejected by the system with shorter windows, 24 seconds was due to an interview with Bill Clinton being re-broadcast, and the rest was either silence, theme-music or the pre-recorded introduction to the show. A more accurate value for the proportion of news information erroneously discarded is therefore about 0.1%.

To improve non-news detection, portions of audio which occur between two repeated sections of audio and have a duration less than the smoothing time, T_s, are relabelled as non-news. As shown in Table 5, by choosing T_s as 1 minute, the proportion of non-news rejected can be increased to 73.1% with no further loss of news content.

windows	T_s (s)	non-news	news
5s long	0	60.8%	0.49%
1s shift	60	73.1%	0.49%
	75	74.5%	0.77%
10s long	0	65.8%	0.51%
2s shift	60	72.4%	0.51%
	75	73.1%	0.79%

Table 5: News and non-news rejection when smoothing between repeated audio

A similar method for non-news detection was included in the Cambridge Spoken Document Retrieval system for the TREC-8 evaluation[5]. Here audio previously broadcast was searched while processing the current broadcast. The system allowed 8.4% of the audio data to be discarded before speech processing, with 97.8% accuracy when compared to manually generated news/non-news labels. As well as reducing the overall computational effort of the system, removing non-news portions had a beneficial effect on information retrieval performance on the transcribed document collection and produced an increase in the mean average precision of the system of 1.2% relative[6].

6. CONCLUSIONS

This paper has described an accurate and fast method of finding exact matches in audio data using a covariance representation of frames of cepstral coefficients and the arithmetic harmonic sphericity distance measure.

The technique has shown to be effective at finding small sections of cue audio and has been applied to automatically locating the end of broadcast news shows and finding non-news events such as commercials, in a database of broadcasts. In this case, some broadcast structure is revealed and the removal of unwanted non-news portions from subsequently transcribed shows leads to a reduction in processing required and improvement in retrieval performance.

ACKNOWLEDGEMENTS

This work is in part funded by an EPSRC grant reference GR/L49611. The idea of using audio repetitions to label commercials was suggested by Tony Robinson.

7. REFERENCES

[1] F. Bimbot & L. Mathan *Text-Free Speaker Recognition using an Arithmetic Harmonic Sphericity Measure.* Proc. Eurospeech '93, pp. 169-172, 1993

[2] J.T. Foote *An Overview of Audio Information Retrieval* Multimedia Systems, Vol. 7, No. 1, pp. 2-10, Jan 1999

[3] J.T. Foote *Content-Based Retrieval of Music and Audio* Multimedia storage and Archiving Systems II, Proc. SPIE, Vol. 3229, pp. 138-147, 1997

[4] J. Garofolo, E. Voorhees, C. Auzanne, V. Stanford & B. Lund *1998 TREC-7 Spoken Document Retrieval Track Overview and Results* Proc. TREC-7, NIST SP 500-242, pp. 79-90, 1999

[5] J. Garofolo, C. Auzanne & E. Voorhees *1999 TREC-8 Spoken Document Retrieval Track: Overview, Results and Analyses* To appear in Proc. TREC-8

[6] S.E. Johnson, P. Jourlin, K. Spärck Jones & P.C. Woodland *Spoken Document Retrieval for TREC-8 at Cambridge University* To appear in Proc. TREC-8

[7] S.E. Johnson *Who Spoke When? - Automatic Segmentation and Clustering for Determining Speaker Turns* Proc. Eurospeech'99, pp. 2211-2214, 1999

[8] K. Kashino, G. Smith & H. Murase *Time-Series Active Search for Quick Retrieval of Audio and Video* Proc. ICASSP'99, pp. 2993-2996, 1999

[9] E. Wold, T. Blum, D. Keslar & J. Wheaton *Content-based classification, search and retrieval of audio* IEEE Multimedia, pp. 27-36, Fall 1996

[10] P.C. Woodland, M.J.F. Gales, D. Pye & S.J. Young The Development of the 1996 Broadcast News Transcription System Proc. DARPA Speech Recognition Workshop, pp. 73-78, 1997.