An Audio Retrieval Algorithm Based on Audio Shot and Inverted Index

Xueyuan Zhang and Qianhua He

***Abstract*—An efficient audio indexing and retrieval algorithm is proposed to locate similar audio segments in the database. A new boundary detection technique based on audio shot is proposed for audio segmentation. Subsequently, a new method is employed to convert the audio shot sequence to audio word sequence, which utilizes a self-learning audio shot dictionary. We also borrow the idea of inverted file from text retrieval to locate candidates efficiently. Furthermore, a similarity measure combining content and temporal order matching is proposed. Experiment results show a retrieval precision of 94.70% within an average response time of 6.344 seconds.**

***Index Terms*—Audio retrieval, audio word, inverted file, temporal similarity.**

I. Introduction

Although audio-based video analysis does not attract as much attention as image-based methods, audio component plays an important role in video. For example, sound film is far more attractive than silent movie because of dub and incidental music. Also, audio approaches generally require less storage of features and computational cost than visual methods. Sometimes audio may be the only available information such as broadcast news, music, telephone conversation recording and meeting recording. In this paper, we focus on audio based indexing and retrieval method.

Most of the existing audio retrieval systems are based on semantic indexing, which automatically transfer low-level audio features into high-level content description according to models. However, this approach relies on supervised model training and it is difficult to bridge the semantic gap due to content variety. An alternative approach is example based audio retrieval method [1] which utilizes audio clip submitted by user as the query. It has a wide range of applications such as unauthorized movie or music detection, violent or erotic clip tracing and duplicates elimination in database management system. Also, a user can easily create a query with materials or simply use an existed audio segment, with which he can retrieve similar movie, music or video scenes and materials.

Little research effort has been invested so far in building data structure for consecutive audio stream. Most research is focused on searching manually indexed clips. However, such manual annotation is subjective, tedious and time-consuming. We propose the concept of audio shot for audio segmentation. Besides, traditional methods [2] apply a sliding window to search for matching segments, which is very time-consuming. To filter out the irrelative parts efficiently, we apply the inverted file that is popular in text retrieval to audio content indexing. The relative parts are cut out and ranked according to their similarity with the query. Also, we define a similarity measure as a combination of content matching and temporal order matching.

The organization of the paper is as follows. The system framework is described in section II. Section III addresses the audio indexing. Audio retrieval is described in section IV. Section V shows experiments and the results. Discussion and conclusion are in section VI.

Manuscript received November 22, 2012; revised January 8, 2013. The work in this paper was supported by the National Natural Science Foundation of China (No. 60972132, No. 61101160)

The authors are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510640, China (e-mail: zhang.xueyuan@mail.scut.edu.cn, eeqhhe@scut.edu.cn)

II. Framework

Fig. 1 shows the proposed example-based audio indexing and retrieval framework, which is mainly composed of three modules, namely dictionary training, indexing and retrieval.

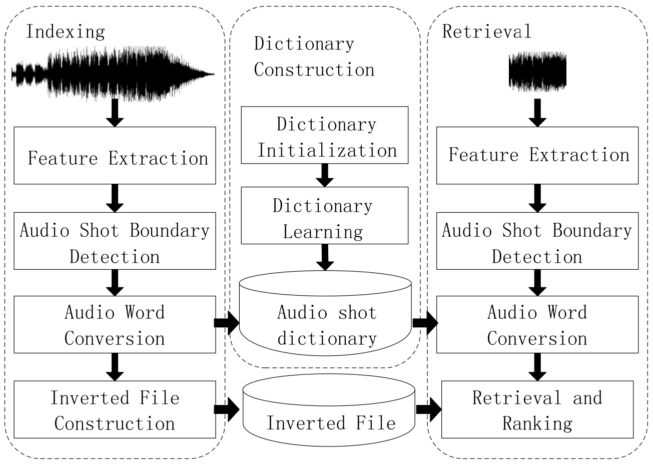

Fig. 1. Audio indexing and retrieval system framework

In order to convert the audio into its content representation, a dictionary is precisely constructed in a semi-supervised way, which comprises two steps: supervised initializing and self-learning updating.

The indexing module aims to construct a data structure for efficient searching. Audio features are segmented into short units called audio shots. The resulted sequences are converted according to the pre-trained dictionary to produce audio word sequences. Then an inverted file is constructed as index and the retrieval process is no longer based on traversing the entire database but scanning the inverted index.

During the retrieval phase, the first three steps are the same as the ones in indexing phase where the submitted query is also converted into an audio word sequence. In the retrieval step, similar audio segments are detected and located using the inverted file. Subsequently, the candidates are ranked based on similarity measures in the ranking step.

The framework is based on audio word that is used to represent the audio content and inverted file that is famous in text retrieval engine for fast full text search.

III. Audio Indexing

A. Audio Word Dictionary Training

A dictionary, which is employed to map the features to audio words, is trained in a semi-supervised way.

To initialize the dictionary, 170 audio effect categories including 6674 samples are collected. For each category, a GMM (Gaussian Mixture Model) with 3 components is trained using EM (Expectation Maximization) algorithm on the corresponding set of observations [3]. The 510 mean vectors are regarded as the entries in the initial dictionary.

To better describe the data, the dictionary is trained in a self-learning manner. A data set, which contains 210 hours audio data extracted from movies, TV programs and broadcast, is used to update the dictionary using a dynamic clustering algorithm Nearest Neighbour Clustering [4].

B. Indexing Process

Seven frequently used audio features are concatenated into a 15-dimensional super-vector, including short-time energy, zero-crossing, spectral energy, 4-dimensional sub-band energy ratio, brightness, bandwidth and 6-dimensional Mel-frequency cepstral coefficients. Each dimension of the feature is normalized to a distribution with zero-mean and unit standard deviation.

In this work, motivated by video shot, we propose a concept of audio shot for audio segmentation. As the fundamental unit to organize a video, a video shot is a consecutive sequence of frames captured by a camera. We define the concept of audio shot as a consecutive sequence of audio frames without major change occurs among them.

According to video processing approaches, shot boundaries are categorized into abrupt and gradual transitions, which are detected by fixed-length and increasing-length windows illustrated in Fig. 2. T^2 distance [5] is employed to calculate the distance between two adjacent data blocks.

$$T^2 = \frac{b(N-b)}{N}(\mu_1-\mu_2)^T \Sigma^{-1}(\mu_1-\mu_2) \tag{1}$$

where N and Σ denote the total window length and covariance matrix respectively, b and μ_1 represent the length and the mean of data in the left data block and μ_2 is the mean of data in the right block. Local peaks of the distance curve beyond a threshold are regarded as shot boundaries and the audio is segmented into audio shots. The mean value of data in each shot is regarded as a representative frame.

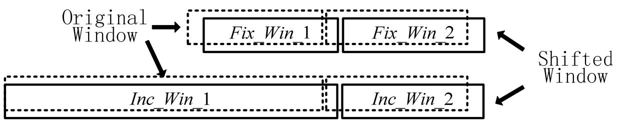

Fig. 2. Fixed-length window and Increasing-length window

Then according to the trained dictionary, the representative frames are converted to audio words by the nearest criterion, which are used as keywords in text retrieval.

Inverted index [6] is the most widely used indexing structure in document retrieval systems. It converts the document-word information into word-document information. An index term list, together with an event list that shows the positions of each index term in all the documents, is included in an inverted file. In our work, the dictionary is used as the index term list. When a document needs to be indexed, the position of each word in the new document is added to the event list of the corresponding index term. A position (i,j) denotes the j-th audio word in the i-th document.

IV. Audio Retrieval

In the example based retrieval, the user submits a clip of audio as the query to retrieve the similar ones from the database. We convert this objective into following four steps: extract features from the query; segment the query into audio shots; convert the audio shots to audio words; generate candidates according to inverted file and ranking. The former three steps are same with indexing phase. In this section we introduce the final step.

Given a query $que=\{q_1,q_2,\ldots,q_{N_Q}\}$, the index term list in the inverted file is searched with each word in *que*. All positions in the corresponding event list of matching term are regarded as *hitting positions*. Assuming candidate and query have the same length, a *hitting position* (i,j) hit by query word q_n generates a candidate: the clip from the j-$(n$-1)-th audio word to the j+$(N_Q$-$n)$-th audio word in the i-th document. All segments that share at least one common audio word with the query are cut out from the database as candidates.

Then, all candidates are ranked according to their similarity with the query in a combination of audio word hitting ratio and temporal order. The hitting ratio, which represents the content similarity between the query and a candidate, denoted as *can,* is defined as,

$$hitting\_ratio = \frac{\#\{can \cap que\}}{\#\{can\}} \tag{2}$$

For temporal order matching, position similarity is employed. $hit=can\cap que$ denotes all the audio words in the intersection of candidate and query. We calculate the temporal order similarity by the average common word distance.

$$order\_ratio = \frac{\sum_{\forall i, hit_i \in hit} |hit\_can_i - hit\_que_i|}{\#\{hit\}} \tag{3}$$

where hit_i is the i-th element in *hit*, hit_can_i and hit_que_i

are the positions of hit_i in candidate and query respectively. The *order_ratio* represents the average position dissimilarity of common audio words. A smaller value indicates a more similar sequence temporal order.

The overall similarity is the combination of *hitting_ratio* and *order_ratio,* which is scaled to interval [0,1] by the exp(•) function. A weighted sum is calculated by (4), where w_1 and w_2 are two adjustable weights with the sum of 1. The contribution of content matching and temporal order matching can be adjusted by its corresponding weight, both of which are generally set to 0.5.

$$Sim(can, que) = w_1 * hitting\_ratio + w_2 * e^{-order\_ratio} \tag{4}$$

Finally, candidates are ranked by similarity. A higher value indicates a higher similarity and the candidate is in a higher-ranked position.

V. Experiment Result

We compare the proposed algorithm with a conventional method called the time-series active search (TAS) [2], which models the distribution of feature vectors with histograms and accelerates the search with dynamic skip width.

A server with 8 CPUs and 32.0GB RAM is used to conduct the experiment with Matlab 2010. The data corpus is composed of 197 Emmy Award-winning series and Oscar winning films with a total length of 270 hours. The data distribution is listed in Table I.

TABLE I: Data Distribution

Video Type	War Movie	Action Movie	Disaster Film	Documentary	Musical	Sitcom
Video Number	15	19	22	38	23	80
Total length (h)	52.6	45.9	39.2	47.6	43.9	40.8

Segments with different lengths are randomly cut out from the database as queries. The positions of queries are recorded as labels to evaluate the effectiveness of the proposed method. A *matching* is defined as the first retrieved result whose overlap with the query label is more than 50%. In the retrieval application, the most concerned problem includes: if the top 1 returned the result is the matching; if the top 10 returned results contain the matching; system response time. We evaluate the performance of the proposed algorithm according to these three items:

- Precision in Top 1: the proportion of queries by which the first returned result is the matching.
- Precision in Top 10: the proportion of queries by which the matching is contained in the first ten returned result.
- Response time: the average response time across queries. Generally the indexing is finished before retrieval. Therefore the indexing consumption is not included. Only the computation time during retrieval phase is shown in this research.

We also evaluate the system performance with queries with different length. The algorithm does not constrain the query length but generally a user will not upload a too long or short. In this research, we evaluate four query lengths: 5, 10, 15 and 20 seconds. All the queries are randomly cut from the database without any assumption about their beginning points or contents.

Table II shows the precision in top 1 and top 10 and the average response time with 20000 queries for each query length.

TABLE II: Retrieval Performance by Queries of Different Lengths

Query length (seconds)		5	10	15	20
The conventional method	Precision in Top 1	80.12%	86.20%	90.12%	92.98%
	Precision in Top 10	86.71%	91.35%	94.03%	95.15%
	Response time (seconds)	55.396	55.480	55.915	56.068
The proposed method	Precision in Top 1	82.65%	87.33%	91.18%	94.70%
	Precision in Top 10	86.83%	90.51%	94.14%	96.60%
	Response time (seconds)	2.150	3.651	5.025	6.344

From the results, it can be observed that the proposed method improves the Top 1 retrieval performance by 3.15%, 1.31%, 1.17% and 1.84%, and shows comparable precision in Top 10 results, while the response time is only 3.88%, 6.58%, 8.99% and 11.31% of the conventional method with different query lengths. Even though the response time increases with longer query, the reduction is still significant compared with the traditional method. The proposed method reduces the response time dramatically and at the same time maintains the retrieval performance.

VI. Conclusion

The proposed algorithm shows a complete audio indexing and retrieval framework. Audio content is segmented according to its content and converted to audio word sequence according to a semi-supervised audio dictionary. During the retrieval phase, by involving both content and temporal order matching, the retrieval performance is improved. The experimental results show the proposed method is 10-fold faster than TAS. The reason for computational time reduction is the use of audio word and inverted-file indexing structure. The audio word enables a brief representation of audio content and the inverted-file, widely used in text retrieval, avoids linear searching and maintains the precise retrieval results. The framework slightly increases the retrieval performance with the utilization of the semi-supervised audio dictionary, where the entries are updated to generate an appropriate set to precisely represent the audio content, and the temporal similarity measurement, which insures a good retrieval performance. The proposed example-based audio retrieval algorithm simultaneously achieves a high precision and short response time.

REFERENCES

[1] H. Heryanto, S. Akbar, and B. Sitohang, "Direct access in content-based audio information retrieval: A state of the art and challenges," in *Proc. 2011 International Conference on Electrical Engineering and Informatics*. Bandung, Indonesia, July 17-19, 2011.

[2] K. Kashino., T. Kurozumi, and H. Murase, "A quick search method for audio and video signals based on histogram pruning," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 348-357, Sept. 2003.

[3] H. Watanabe, S. Muramatsu, and H. Kikuchi, "Interval calculation of EM algorithm for GMM parameter estimation," in *ISCAS*, Paris, France, May 2010, pp. 2686-2689

[4] S. Bubeck and U. Luxburg, "Nearest neighbor clustering: a baseline method for consistent clustering with arbitrary objective functions," *JMLR*, vol. 10, pp.657-698, 2009.

[5] B. Zhou and J. H. L. Hansen, "Efficient audio stream segmentation via the combined T2 statistic and Bayesian information criterion," *IEEE Trans. Audio Speech Lang. Process.,* vol. 13, no. 4, pp. 467- 474, July 2005.

[6] B. Croft., D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*, Addison Wesley, Feb. 2009, pp.131-141.

Xueyuan Zhang was born in Shijizhuang, Hebei Province, China on 5 Nov 1987. He received the B.E. degree in Information Engineering from South China University of Technology (SCUT) in 2009. Since 2010, he was pursuing the Ph.D. degree in Information and Communication Engineering from SCUT supervised by professor He Qian-hua..

He is currently a visiting student to Center for Speech, Language and Brain, Department of Experimental Psychology in University of Cambridge, United Kingdom. His research interests include speech recognition, speaker diarization and content based audio indexing and retrieval. Currently, he is involved in research on investigating the cognition process of the brain.

Qianhua He was born in Shaodong, Hunan Province, China on 2 Feb. 1965. He received the B. S. Degree in physics from Hunan Normal University in 1987, the M. S. Degree in medical instrument engineering from Xi'an Jiaotong University in 1990, and the Ph. D degree in communication engineering from South China University of Technology, in 1993. Since 1993, he has been at the Institute of Radio and Auto-control of South China University of Technology.

His research interests include speech recognition, speaker verification security, optimal algorithm (such as genetic algorithm and neural networks), embedded system design. From 1994 to 2001, he worked with the department of computer science, City University of Hong Kong for about 3 years in 4 periods. From 2007.11 to 2008.10, he worked with University of Washington in Seattle as a visiting scholar.