FINGERPRINTING TO IDENTIFY REPEATED SOUND EVENTS IN LONG-DURATION PERSONAL AUDIO RECORDINGS

James P. Ogle and Daniel P.W. Ellis

LabROSA, Dept. of Electrical Engineering
Columbia University, New York NY 10027 USA
jpo2101@columbia.edu, dpwe@ee.columbia.edu

ABSTRACT

Body-worn solid-state audio recorders can easily and cheaply capture the bearer's entire acoustic environment throughout the day; we refer to such recordings as "personal audio". Extracting useful information, and providing access and navigation tools for this data is a challenge; in this paper we investigate the use of an audio fingerprinting technique, originally developed for identifying music recordings corrupted by noise, as a tool to rapidly identify recurrent sound events within long (multi-day) recordings. The fingerprinting technique is based on energy peaks in time-frequency, largely removing framing issues and making it intrinsically robust to background noise levels. We show that the technique is very effective at identifying exact repetitions of structured sound (such as jingles and electronic telephone rings) but is unable to find repeats of more 'organic' sound events such as garage door openings.

Index Terms— Search methods, Database searching, Acoustic signal analysis, Fingerprint identification.

1. INTRODUCTION

Recent technology advances in digital storage combined with new audio compression techniques have made audio recording easy and affordable. Many compact MP3 players with gigabyte storage capacity and built-in microphones are available on the market today for around $100. These can be worn on a belt or on a lanyard around the neck allowing relatively unobtrusive recording. This makes it feasible to record virtually all one hears creating an audio history. The collection of all of these recordings can form a type of personal audio archive. Unfortunately, the long duration of these types of recordings make it difficult to find a particular event of interest. This quickly limits the practical uses for such an archive of comprehensive personal audio recordings.

To build practical applications that can take advantage of a personal audio archive, processing techniques to make it easy to find a specific sound event, episode, or class of events will be needed. Early research in this area has started to explore these types of tools even while the ultimate application of the data is still unclear. Gemmell *et al.* have a project to capture a wide array of data encountered in ones life including audio and they are working on evaluating methods to organize and manage the results [4]. Ellis and Lee have presented features and methods for possible segmentation a classification in this domain [2]. Common to this research is the realization that data management tools are critical.

The work described herein focuses on one aspect of data management in this domain, the similarity search. At one level or another, the ability to search the archive for a sound event of interest will be critical to potential applications. A technique to identify all recurring events within a large audio corpus rapidly and comprehensively would be useful within an unsupervised process to discover structure in these recordings. Similarity searches have proven useful in the 'production audio' domain for a variety of applications. Wang has documented Shazam Electronics commercialized audio search algorithm used to identify songs within a large database from a noisy example query recorded by a cell phone [10]. Kashino *et al.* proposed a system to identify repeating multimedia segments from long duration streams using vector quantized low-level feature vectors [8]. Herley outlined a number of potential applications for identifying repeating objects in multimedia streams and proposed using a low-dimension fingerprint and statistical search to accomplish the task [6].

This paper explores whether these types of similarity searches can be applied to the personal audio recording domain to find repeating sound events. Specifically, an implementation based on the sparse landmark fingerprint and hashing technique of [10] is adapted and evaluated for search and retrieval of general sound events typical of environmental recordings. Whereas many audio fingerprinting algorithms project the entire audio signal onto some compact feature representation, then accommodate distortions or noise by the use of advantageous features [7, 5, 1], sparse landmarks take a different approach in which identification is based only on a very small number of prominent energy peaks, making recognition largely insensitive to overlapping simultaneous sounds – until the prominent peaks are overwhelmed. This makes it particularly interesting for our task.

The rest of this paper is organized as follows: Section 2 provides an overview of the algorithm including discussion of the fingerprint technique and search mechanism. Section 3 presents the experiments conducted to evaluate the algorithm and the results. Section 4 provides conclusions and suggestions for further study.

2. ALGORITHM

The basic principle behind the algorithm is to extract low-level features from the audio signal to form a low-dimension representation that can then be efficiently and accurately compared for a similarity match. This general technique is often referred to as fingerprinting and there has been much research in this area. The particular fingerprints we employ are based on 'sparse landmarks' – compact local energy peaks that are intrinsically robust to background noise at any but the highest levels, and can even provide discrimination in sound mixtures; this is particularly important for personal audio recordings where recording conditions and background noise may be very variable.

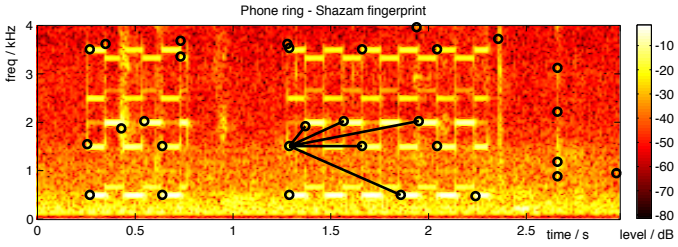

Fig. 1. Landmark points selected by the algorithm for a telephone ring.

2.1. Fingerprint

The basic fingerprint is based on the music identification system of [10]. Starting from a short-time Fourier transform magnitude (i.e. the spectrogram), landmarks are identified as the onsets of narrow spectral peaks, described only by their time and frequency coordinates. The spectrogram is coarsely divided in time and frequency to create equal spaced zones. A fixed number of peaks are chosen from each zone as landmark points. This method ensures a roughly uniform distribution of landmarks in time-frequency. Figure 1 shows an example spectrogram and the associated landmark points.

The next step is to form hashes from combinations of landmark points. Each landmark is selected as an anchor point and paired with nearby landmarks from within a target zone. The target zones include a fixed number of points so each landmark point generates a fixed number of pairs. A hash is then formed of the two frequencies and the time difference between them. We used a fan-out of 9 pairs per landmark, resulting in a hash density of 162 hashes/second. Each hash value is stored in a database along with the audio file in which it was found, and the absolute time at which it occurred.

The power of this representation is in improved robustness to noise, improved discrimination, and search efficiency. By choosing spectral maxima as landmark points, the algorithm naturally filters out background noise. By using pairs of frequency points rather than single points, the representation contains information regarding the time-frequency structure of the sound which aids in discrimination. The hashes can be stored compactly and hash table constructs can be used to enable very efficient searches. The algorithm employs 64 frequency bins and the time delta is quantized to 6 bits resulting in an 18 bit hash, or 262,143 distinct hash values. This finite hash space facilitates efficient search and matching.

2.2. Search

Once a series of audio files have been fingerprinted, it is possible to perform similarity searches in the fingerprint domain. A query signal is fingerprinted and the resulting hashes are compared against the archive hashes. If a hash match is found, the file time offset for the query hash and the file time offset for the archive hash are stored as a pair. After all of the hashes of the query signal have been compared against the archive hashes, the file offset pairs from matching hashes are subtracted resulting in a series of time differences. These deltas can then be clustered into bins to form a histogram of time offsets. Two similar sound events, containing similar or identical spectral peaks at the same relative timing, will generate many common hashes at consistent relative timing; note that it is not necessary for all the same hashes to be generated, only that enough common hashes are generated to differentiate with chance hash collisions. A statistical threshold can be set to identify the peak in the histogram of common hash counts and thus detect a match; we used a constant factor relative to the mean histogram bin value as the threshold. As described in section 3, the appropriate threshold for a match is dependent on the type of sound event and the desired tradeoff between precision and recall.

The result of this process is shown in figure 2, a kind of self-similarity matrix [3] or, in the terminology of chaotic systems, recurrence plot [9]. An autonomous repeat event search using this technique was performed on a 30 min recording containing two instances of song A at 400 s and 1500 s, one instance of song B at 1000 s, and two telephone rings at 600 s and 1100 s. A sliding 2 s window was applied to the signal and used as a query. The horizontal axis represents the time of the query window. The vertical axis represents the archive time. The intensity of the image at any given point represents the number of hash hits for that query. The high-intensity diagonal is the "self match" when the query signal passed over itself in the recording resulting in a very high number of hash hits. Repeated events are identified by peaks in the off diagonal elements. The match of song A to its repeat can be seen at (400,1500) and at (1500,400). The increase in intensity for song A around 1000 s was actually song B; however, this increase in intensity was below the detection threshold for a match.

The hash representation allows a tremendous improvement in search efficiency over methods that requiring a linear search. Hashes, offsets, and file identifiers are stored in an archive fingerprint file. Each hash is recorded along with its time location within the archive, meaning that it is not necessary to preserve the time relationship between these records within the fingerprint file, unlike the fingerprints used in linear time searches [8, 6]. For search efficiency the fingerprint file is sorted by hash value, meaning that all hashes that could potentially be related to similar events are grouped together. This representation allows a hash index file to be built that stores the starting location of the each hash grouping in the fingerprint archive file as well as the number of instances of that hash. This allows the search mechanism to directly index into the fingerprint archive file based on the query hash value to immediately find all instances of that hash in the archive. The result is constant-time hash searches independent of the length of the archive file. There are no unnecessary comparison calculations to determine whether a record should be included in the offset calculations described above.

In a linear time search, fingerprint elements are stored in time-order. The linear search algorithm must scan the file in order checking each fingerprint entry to identify potential matches. While optimizations can be introduced to avoid an exhaustive search, the linear search algorithm can not avoid unnecessary comparisons. Using the hashing search technique presented here, the locations of all relevant fingerprint elements are given in the index file. Because there are a fixed number of hashes, the index file size is fixed. In addition, because of the hash space compression, the size of this file is small at approximately 2 MB. Thus the index file adds very little in storage overhead.

3. EXPERIMENTS

While the ultimate use of a similarity search in the personal audio archive domain may not be immediately obvious, certain characteristics will certainly be required in any application. Recall and precision are typical metrics for these types of searches. In addition, search speed and efficiency will be an important factor in practical applications. Finally, due to the fact that these types of recordings are captured in our natural environment with many different levels of background audio, robustness in noise is an important attribute.

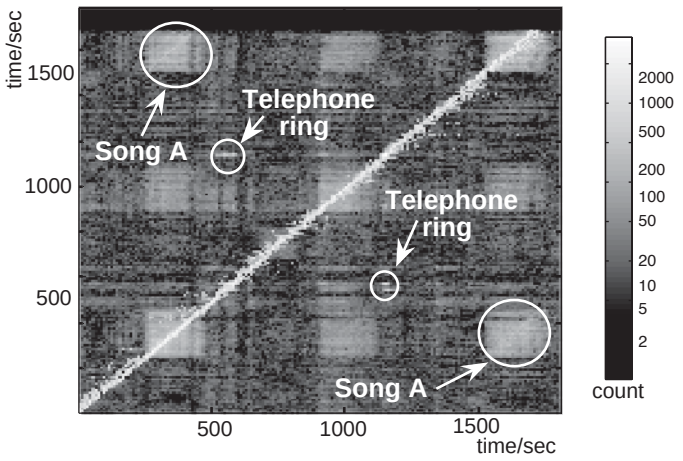

Fig. 2. Exhaustive search for repeating events for a thirty minute recording, showing the histogram of common hashes between each 2 s window. Hash hit counts are represented in log-scaled intensity.

This section explores how the technique performed in these areas.

As mentioned in the introduction, the work associated with this paper was focused on identifying sound events from environmental recordings typical of those which might be found in a personal audio archive. Our world provides an infinite number of sound events to explore. To narrow the scope of the research to a manageable size, the types of sounds examined were limited to three categories: production audio – which included studio recorded music and radio broadcasts; alert sounds – such as telephone rings and door bells; and organic sounds such as doors closures and garage door openings.

3.1. Data Set

The data set included 40 hours of mostly unscripted recordings. These were captured during the first authors typical work day using a compact MP3 recorder at a bit rate of 64 kbps. This was deemed to be consistent with the typical usage of a personal audio archive. The resulting audio files naturally included repeated occurrences of all the sound events in the chosen categories in a variety of environments. In addition, repeated events such as telephone rings and playback of music were deliberately inserted to provide known repeats events for algorithm testing. A single channel of the resulting stereo signal was downsampled to 11025 kHz and processed.

Because of the long duration of these recordings and the manner in which they are obtained, establishing ground truth is problematic. Manually identifying repeating events over many hours of recordings is tedious and error-prone. While deliberately repeated events can be carefully scripted and documented, the quantity of such events within the 40 hour dataset was limited by practical considerations. Thirty songs were introduced throughout the recordings and 45 telephone rings were also added. Other events such as door closures and beeps were introduced with 10 occurrences each.

3.2. General Search Results

The basic mechanics of evaluation was to record query sounds or pull query sounds from the archive, fingerprint the query signal, and search for a match in a long duration archive that had been fingerprinted and indexed in a preprocessing step. Both query and archive audio were captured using the same recorder and subjected to the same mp3 compression.

Table 1. Search metrics for the three sound event categories.

Category	Recall		Precision	
Production Audio	29/30	97%	29/34	85%
Alert Sounds	45/65	69%	45/45	100%
Organic Sounds	0/20	0%	0/20	0%

As can be seen in Table 1 recall and precision varied between the different sound categories. Very good results were generally observed with production audio and certain types of alert sounds while the algorithm performed very poorly with other alert sounds and in the organic sound category.

Production audio was limited to music recorded either from CD playback or television or radio broadcast. Speech and other non- music sounds recorded from production audio sources were lumped into the organic sounds category. Natural background noise was present during recording and its nature and level varied. A search result was considered positive if it contained any portion of the correct song. False positives for music were often encountered in instrumental portions of songs that did in fact contain similar musical content. Many of these false positives could be removed by adjusting the matching threshold.

Performance in the alert sound category was mixed. This category consisted of a variety of telephone rings, door bells, and other electronic beeps. Consistent sounds with some spectral breadth such as telephone rings resulted in statistically significant matches with no false positives. However, single tone beep events were not detected by the algorithm. This is due to the nature of the algorithms hash formation. For single tone sounds there is only one frequency component and thus the majority of the hashes generated for these sounds are combinations of the frequency components of whatever noise happens to be present at the time.

The algorithm does not perform well with organic sounds – the unstructured noise and speech that make up much of our aural world. Human listeners will recognize the similarity between, say, different door closure sounds, but this algorithm will not. Sounds such as impact transients, machinery, and speech tend to vary in spectral and temporal structure in each occurrence and thus they do not generate exactly reproducible hashes. The algorithms failure in this category is not surprising since these types of inconsistent sounds are what the algorithms original implementation was designed to filter out. Unfortunately, one can imagine many instances where identification of repeated events of these types of sounds might be valuable to practical applications in the personal audio domain.

When considering search metrics, it is important to note that the algorithm easily allows a user to adjust the matching threshold to make tradeoffs between recall and precision. Results here were recorded using a set matching threshold to provide a general comparison across categories. In practical use it is likely the matching threshold would be adjusted on the fly to provide the desired level of discrimination.

3.3. Search Speed

An important consideration in evaluating any search mechanism in the personal audio archive domain is speed. As described in section 2.2, the method presented here takes advantage of the fingerprint representation so that a time series search through the archive is not

Table 2. Average search speed for 5 s query on various archive lengths.

Archive Length (min)	60	120	180	390
Search time (ms)	21	31	37	131

Table 3. Performance in noise. Search query was a 2 s telephone ring. No false positives were observed.

SNR / dB	3	-3	-9	-15	-21
Recall	100%	89%	89%	89%	56%

necessary. The hashing technique allows direct indexing into the fingerprint location of interest. Thus searches for matching hashes are performed in constant time. Overall search time is then dominated by the time to calculate the time offsets for all the matching hashes and construct the histogram of offsets. These calculations add a dependency on archive length since the likelihood of finding a hash match increases with archive length and thus the time for offset calculation increases. Experiments found that this time dependency is much less than that for time-series searches though. Some average search times for a five second query on archives of various lengths are presented in table 2. Note the significant increase in search speed for the archive length of 6 h; we believe this is a byproduct of the memory constraints of the PC used for evaluation. The time increase occurs when the archive file no longer can be stored in main memory. Disk access for caching then becomes a significant factor in performance. This shortcoming needs to be addressed in future study.

3.4. Performance in Noise

For environmental recordings such as these, noise robustness is important. All the data was subject to various levels of background noise. However, it is difficult to quantify general noise performance due to the randomness of the natural environment. To provide a more quantitative assessment of noise robustness, an set of experiments were run where white gaussian noise was artificially added to the query signal. Table 3 shows the impact of noise on the recall rate for a telephone ring query. As can be seen the results are quite impressive even for very low SNR levels.

For production audio the results were not quite as impressive. Recall and precision were unaffected down to an SNR of 0dB. However, at SNR levels of -1dB and below, it was necessary to adjust the matching threshold to detect repeats which in turn increased the number of false positives. The amount of adjustment varied from query to query and thus no generalized recall and precision metrics are given here for these experiments.

Robustness in noise was also observed through experiments that explored identifying sounds in audio mixtures where multiple sources were recorded simultaneously. Exploiting this as a means for sound separation is a possible area for future study.

4. CONCLUSIONS

In this paper, we have evaluated a method to detect repeating sound events in personal audio recordings. Experiments have shown the method performs reasonably well for a variety of sound events that are common in environmental recordings. Search speed is impressive and eliminates much of the dependency on archive length that is seen in linear time series approaches. The methods efficiency and accuracy in production audio suggests possible application to broadcast monitoring for copyright or market analysis. Performance in noise including mixtures was found to be very good suggesting robust searches as well as possible other applications in sound separation. Further study needs to explore more efficient static storage structures to overcome the run-time performance penalty seen when the archive file length exceeds dynamic memory capacity.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) under Grant No. IIS-0238301. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. Additional support was provided through the International Computer Science Institute from the EU project AMI.

6. REFERENCES

[1] C. J. C. Burges, J. C. Platt, and S. Jana. Distortion discriminant analysis for audio fingerprinting. *Speech and Audio Processing, IEEE Transactions on*, 11(3):165–174, 2003.

[2] D. P. W. Ellis and K. Lee. Accessing minimal-impact personal audio archives. *IEEE MultiMedia Magazine*, 13(4):30–38, 2006.

[3] J. Foote. Visualizing music and audio using self-similarity. In *Proce. ACM international conference on Multimedia*, pages 77–80, Orlando, FL, 1999. ACM Press.

[4] J. Gemmell, G. Bell, and R. Lueder. MyLifeBits: a personal database for everything. *Communications of the ACM*, 49(1):88–95, Jan 2006.

[5] J. Haitsma, T. Kalker, and J. Oostveen. An efficient database search strategy for audio fingerprinting. In *IEEE Workshop on Multimedia Signal Processing*, pages 178–181. St. Thomas, USVI, 2002.

[6] C. Herley. ARGOS: Automatically extracting repeating objects from multimedia streams. *IEEE Tr. Multimedia*, 8(1):115–129, Feb 2006.

[7] J. Herre, E. Allamanche, and O. Hellmuth. Robust matching of audio signals using spectral flatness features. In *Proc. IEEE Workshop on Apps. of Sig. Proc. to Audio and Acous.*, pages 127–130, Mohonk NY, 2001.

[8] K. Kashino, T. Kurozumi, and H. Murase. A quick search method for audio and video signals based on histogram pruning. *IEEE Tr. Multimedia*, 5(3):348–357, Sep 2003.

[9] N. Marwan, M. Carmen Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, In Press, Corrected Proof:–, 2007.

[10] A. Wang. The Shazam music recognition service. *Communications of the ACM*, 49(8):44–48, Aug 2006.