FINDING SIMILAR ACOUSTIC EVENTS USING MATCHING PURSUIT AND LOCALITY-SENSITIVE HASHING

Courtenay Cotton and Daniel P. W. Ellis

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

ABSTRACT

There are many applications for the ability to find repetitions of perceptually similar sound events in generic audio recordings. We explore the use of matching pursuit (MP) derived features to identify repeated patterns that characterize distinct acoustic events. We use locality-sensitive hashing (LSH) to efficiently search for similar items. We describe a method for detecting repetitions of events, and demonstrate performance on real data.

Index Terms— Acoustic signal analysis, database searching

1. INTRODUCTION

There are many examples of sound events which may be heard multiple times in the same recording, or across different recordings. These are easily identifiable to a listener as instances of the same sound event, although they may not be exact repetitions at the waveform level. We define an event as any short-term, perceptually distinct occurence, e.g. a door knock. The ability to identify recurrences of perceptually similar events has applications in a number of audio recognition and classification tasks. This work was motivated specifically by the desire to relate repeated audio events with the visual source of the sound, as in a video.

Our goal is to identify characteristic patterns that can be used to search for the presence of an event, i.e. identifying a kind of fingerprint for the events. There are two main challenges to this task: First, we must find a representation that is sufficiently invariant to differences in event instances, and to context such as background sounds, to allow repeated events to be matched, yet still captures enough detail of the sound to allow perceptually distinct events to be distinguished. Second, we need a way to efficiently search for these events in very large datasets.

Our approach to the first problem is to use the matching pursuit (MP) algorithm as the basis for our audio event representation. MP [1] is an algorithm for sparse signal decomposition into an over-complete dictionary of basis functions. MP basis functions correspond to concentrated bursts of energy localized in time and frequency, but spanning a range of time-frequency tradeoffs. By allowing the analysis to choose the bandwidth/duration parameter that best fits a feature in the audio – instead of adopting a single, compromise timescale as in the conventional short-time Fourier transform, MP allows us to describe a signal with the atoms that most efficiently explain its structure. The sparseness of this representation makes this approach robust to background noise, since a particular element, representing the most compact local concentration of energy, will experience the least proportional change as the surrounding noise level increases. MP features were proposed for environmental audio classification in [2].

Our work is also inspired by previous work in searching for events using a strict, exact-match fingerprint technique [3]. That algorithm efficiently identified audio excerpts that were repeated in their entirety, such as pieces of music or electronic ring tones, from environmental audio. However, it was not able to identify "organic" sounds (such as the sound of a door closing) where there was nontrivial variation between successive instances of the event. In the current paper, we use a similar representation in terms of time-frequency energy peaks taken in pairs and characterized by their time difference, but instead of an exact hash we use locality-sesitive hashing (LSH) [4], an algorithm that uses the highly efficient constant-time mechanism of hash lookups to find nearest neighbors in feature space instead of only exact matches. LSH has been proposed for matching similar music items e.g. remixes of particular tracks [5].

Section 2 describes our MP representation, section 3 describes how we search for recurring events, section 4 describes our preliminary experiments to illustrate this idea, and we conclude with a discussion of the issues raised and the prospects for unsupervised discovery of repeating acoustic events.

2. MATCHING PURSUIT REPRESENTATION

The basis functions used for MP are Gabor functions, i.e. Gaussian-windowed sinusoids. The Gabor function is evaluated at a range of frequencies covering the available spectrum, scaled in length (trading time resolution for frequency resolution), and translated in time. Each of the resulting functions is called an atom, and the set of atoms is the dictionary, which covers a range of time-frequency localization properties. The length scaling creates long atoms with narrowband frequency resolution, and short atoms (well-localized in time) with wideband frequency coverage. This amounts to a modular STFT representation with analysis windows of variable length. During MP analysis, atoms are selected in a greedy fashion to maximize the energy removed from the signal at each iteration, resulting in a sparse representation. Atoms extracted from the signal are defined by their dictionary parameters (center frequency, length scale, translation) and by parameters the algorithm estimates (amplitude, phase). Here, we use the Matching Pursuit Toolkit [6], an efficient implementation of the algorithm.

This work was supported by the NSF (grant CNS-0716203), the Eastman Kodak company, and EU project AMIDA (via the International Computer Science Institute). 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 sponsors.

The dictionary we use contains atoms at nine length scales, incremented by powers of two. For data sampled at 22.05 kHz, this corresponds to lengths ranging from 1.5 to 372 ms. These are each translated by increments of one eighth of the atom length over the duration of the signal.

2.1. Psychoacoustic Pruning of Atoms

Since MP is a greedy algorithm, the first (highest energy) atom extracted from a given neighborhood is the most locally informative, with subsequent lower energy atoms selected to clean up imperfections in the initial representation; these are often redundant in terms of identifying key time-frequency components.

In order to reduce the size of the representation while retaining the perceptually important elements, we prune the atoms returned by MP with post-processing based on psychoacoustic masking principles [7, 8]. The objective of MP is to reduce the energy of the error residual as much as possible at each stage, but owing to the limitations of human hearing, perceptual prominence may be only weakly related to local energy. In particular, lower energy atoms close in frequency to higher-energy signal may be entirely undetectable by human hearing, and thus need not be represented. A related effect is that of temporal masking, which masks energy close in frequency and occurring shortly before (backward masking) or after (forward masking) a higher-energy signal; the forward masking effect has a longer duration, while the backward masking is typically negligible. A similar approach, which incorporates a psychoacoustic model into the matching pursuit algorithm itself, has been explored in several places, such as [9].

Our implementation creates a masking surface in the time-frequency plane, based on the atoms in the full MP representation. Each atom generates a masking curve at its center frequency with a peak equivalent to its amplitude, which falls off with frequency difference. Additionally we consider this masking curve to persist while decaying for a brief time (around 100 ms), to emulate forward temporal masking.

Atoms with amplitudes that fall below this masking surface are therefore too weak to be perceived in the presence of their stronger neighbors; they can be removed from the representation. This has the effect of only retaining the atoms with the highest perceptual prominence relative to their local time-frequency neighborhood. This pruning emphasizes the most salient atoms, and removes less noticeable ones; it is an important step in reducing the size of the search space and improving the relevance of the atoms as features (since secondary, "cleaning up" atoms are usually removed), as well as reducing the probability of false matches in the search procedure described below.

Figure 1 shows an example audio clip containing two distinct transient events analyzed with MP. A large set of MP atoms (171 in this case) is extracted initially and then pruned with psychoacoustic masking (leaving 76 in the example).

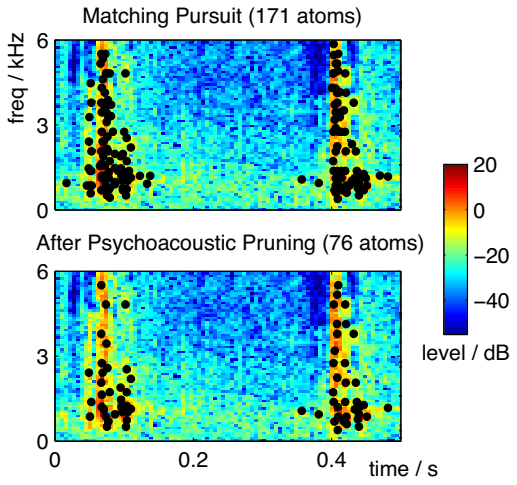

Figure 1: Matching pursuit atoms used to represent audio events, shown as dots at the centers in time-frequency overlaid on the spectrogram, before and after psychoacoustic pruning.

3. PAIR FORMATION AND PATTERN DISCOVERY

We want a way to define a specific type of audio event by the common relationships between its atoms, if any exist. We start with an audio sample containing many separate instances of the same event that we would like to describe, potentially including small variations in the detail the instances. We form pairwise relationships between all pairs of atoms whose centers fall within a relatively short time window of each other. The pairs are characterized by the two center frequencies of the atoms and the time offset between them, creating a three-dimensional feature space. Bandwidth parameters could also be used to refine the space, but were not needed for the current demonstration. By excluding the energy of atoms, the representation becomes robust to variations in level and channel characteristics, provided the sound event energy remains sufficiently prominent to be included in the MP analysis.

The dimensions are normalized, then LSH is performed on the entire set of pairs. LSH makes multiple random normalized projections of data items onto a one-dimensional axis. Items that lie within a certain radius in the original space will be sure to fall within that distance in the projection, whereas distant items have only a small chance of falling close together. The projected values are quantized, and near neighbors will tend to fall into same quantization bin. These quantized sets are quickly recovered at query time through a conventional hash table. By consolidating the results across multiple projections, both chance co-ocurrences due to unlucky projections, and the risk that nearby points will straddle a quantization boundary can be averaged out.

We start with a soundfile that we know or suspect contains multiple instances of some sound event (perhaps mixed in with other, nonrepeating events). We form the MP representation, then store hashes describing all nearby atom-pairs in an LSH database. The database is then queried with every atom-pair hash in turn to identify large clusters of similar pairs i.e. atom-pairs that return large numbers of matches within some radius. Since the LSH queries are constant-time, this entire process takes a time proportional only to the number of atom-pairs, instead of the N^2 time required for exhaustive pairwise comparison. We assume clusters in atom-pair space arise from the repeating events, and we can link pairs into higher-order constellations if they share individual atoms. Thus, we arrive at a set of pairs we can use to recognize future instances of the repeating event. Each acoustic event may be result in dozens of nearby atoms, leading to many local pairings. Detection does not require that *all* atoms and relations be successfully identified; it is frequently sufficient to detect only a small fraction of these "landmarks" to correctly identify a structure.

LSH requires a radius parameter be set to define how close nearest neighbors must be. This is essentially the definition of how "similar" the particular time-frequency structure of the characteristic features of two events must be for them to be considered instances of the same event class. Here the radius is tuned by hand, but this could easily be automated for instance based on an estimate of the true number of event repetitions in this training set.

4. EXPERIMENTS

We test our approach on a 13 second sample from a video soundtrack, containing 38 instances of the sound of horse hooves. This is a real recording, i.e. each hoof sound arises from a distinct physical event. We extract the top 3000 atoms with MP and perform psychoacoustic pruning on the set of atoms, which reduces this number by about half. We use a ± 35 ms time window for pair formation between atoms, which produces around 19,000 pairs.

We use a small LSH radius of 0.085 (in the normalized feature space) to cluster this set of pairs; this radius was tuned to give a relatively small number of hits, such that only very similar, hopefully characteristic pairs will be matched with each other. For this radius, the most common pair pattern had 35 nearest neighbors. We then select patterns with at least 20 nearest neighbors, which yields 25 pair patterns, each of which we infer occurs in at least half of the hoof sound events. Because the events we are describing here are very short in time, most of these pairs appear to be nearly simultaneous (i.e. very little time offset between the two atom locations) although some show a small but consistent time skew between center frequencies.

4.1. Event Detection

To test our algorithm, we mix the signal with a second soundtrack containing speech and general background noise. Listening to the mixtures, as a second signal is added many of the horse hoof events become less audible, especially those that overlap with speech, but those that are audible remain distinctly identifiable; it is this effect that we hope to reproduce. Even as the confusing speech and noise is increased, the atoms and pairs representing audible events should hopefully remain reasonably similar.

Figure 2 shows the percentage of nearly identical atoms retained from the original (pruned) atom represenation of the clean signal, as the mixing proportion changes. Lower SNR corresponds to more of the second, masking signal being added.

We perform our pair extraction process (MP, psychoacoustic pruning, and pair formation) on the mixture at varying SNRs, form atom-pairs and store them in an LSH database, then query this database using our previously saved 25 queries generated from the original clean signal. As an experimental variation, we try several different LSH radii including the "true" radius used to define the characteristic query pairs. At each SNR level, the matches found for each of the 25 queries are scored as true or false positives; each query is scored separately based on the locations of its matches in the original signal. True positives are matches found within a small window (5 ms) near the time of a similar query match in the original signal. False positives are any other matches found in the mixture. Although each of the hoof events usually contains multiple pairs, one pair found within an event is considered sufficient to detect that event.

Figure 3 shows the precision and recall of the system, at three different LSH radii (0.025, 0.085, 0.15). Recall is defined as the number of events (out of 38 total) with at least one match detected. Precision is defined as the number of correct matches found over all queries divided by the sum of all query matches.

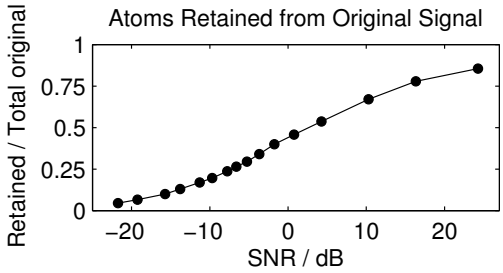

Figure 2: Percentage of atoms from the original signal retained as the noise increases.

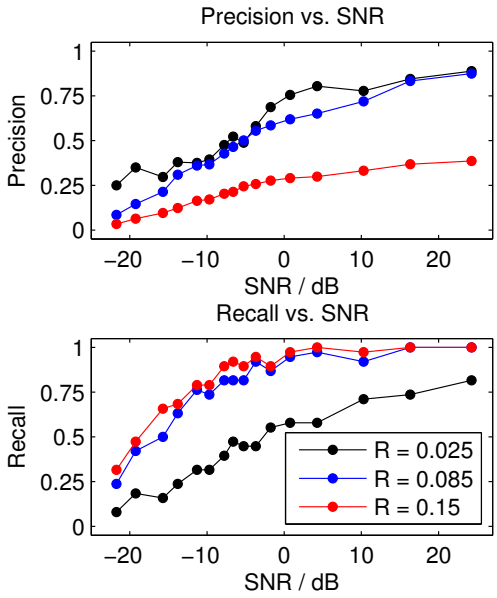

Figure 3: Detection, precision, and recall vs. SNR at three LSH radii.

Figure 4 shows a portion of the original signal, with the locations of query pairs marked; below it is the signal in a mixture, with true matches (black) and false positives (magenta) marked.

5. DISCUSSION

Figure 2 shows that the pruned atom representation of the original events degrades reasonably gracefully as a second signal is added, with almost 25% of the atoms remaining essentially unchanged even at -10 dB SNR. These many MP atoms that stay nearly identical, even in noise, indicate that the atom pair patterns have the potential to form a noise-robust basis for identifying the sound events.

Figure 3 demonstrates that recall is reasonably stable over a range of SNR. This follows from our simple approach of OR-ing together all of the queries to detect the 38 events, with the corresponding negative impact on precision. More complex comparison

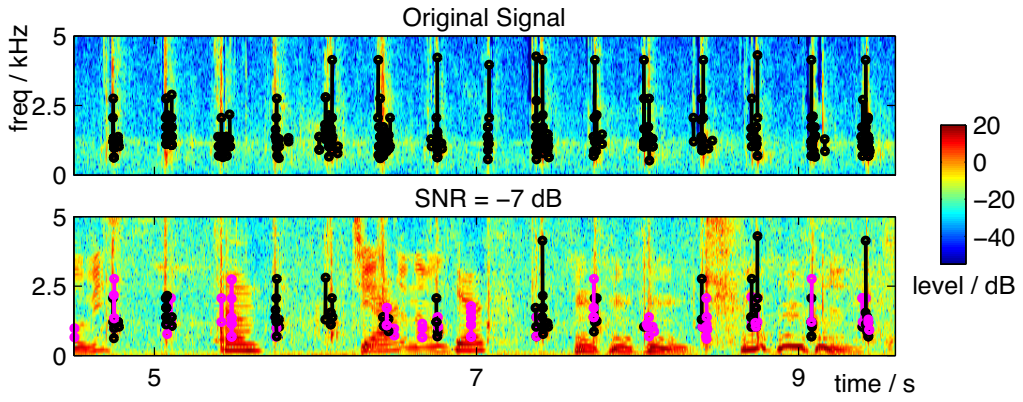

Figure 4: Original query patterns (top), and mixture detection results (bottom). Black pairs are correct detects; magenta are false positives.

of the relations between pairs, such as noting which pairs share end-points in the original data and requiring that this sharing be preserved at recognition time, could potentially improve precision without impacting recall. It is promising, however, that nearly all of the events maintain a similar enough representation so that at least one nearly identical pair persists in each, even at fairly low SNR. This indicates that the atomic representation of the original events is staying relatively constant as the second signal is added.

The tradeoff between precision and recall is seen in the variation with radius. Results at a radius of 0.085 correspond to the radius for which the queries were originally chosen. Choosing a larger radius improves recall, if only slightly, at the cost of precision. A smaller radius will improve precision (slightly), but with a low recall rate.

Figure 4 shows examples of successful and unsuccessful event detection in noise. The top shows all instances of the 25 originally selected patterns. Below, the atom pairs in black are those that have been found nearly identically at low SNR as they were in the original signal. The magenta pairs are false positives. There are also a few events which have been lost entirely due to the presence of noise.

6. CONCLUSIONS

We demonstrate a promising approach for the detection of repeated events in large amounts of audio data. The patterns identified here are robust enough to be useful for event detection, even in the presence of noise. Practically, the main shortcoming is probably low precision, indicating that the patterns are not entirely unique to the event under consideration. However, this can probably be improved upon by tuning parameters such as the atom pruning threshold and LSH radius in both the discovery and detection stages of the algorithm. When selecting queries, we could also consider not just its commonality in that event, but the frequency with which it is found in other generic audio; this would allow us to select patterns more unique to a specific event type and therefore improve precision.

There are several obvious enhancements which could make the algorithm more robust. Pair representations could be made more specific by incorporating other atom parameters into the feature space, such as atom length or amplitude difference between the pair. Pairs often found together could be joined into constellations, producing something closer to a fingerprint (or realistically, a set of potential fingerprints) for an event. A set of common constellations could be stored as potential queries for the detection task. Alternately, the pairs do not necessarily need to be explicitly linked together; they could each form an individual event detector, the results of which could be combined probabilistically for greater stability.

A set of atom pairs or constellations could be identified to define sets of events of different types. As demonstrated here, LSH would allow for efficient searching of any generic audio recording for the presence of patterns matching any of these events, which could be very promising for audio-visual analysis, among other applications.

7. REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, Dec. 1993.

[2] S. Chu, S. Narayanan, and C. Kuo, "Environmental sound recognition using mp-based features," in *Proc. IEEE ICASSP*, 2008.

[3] J. Ogle and D. Ellis, "Fingerprinting to identify repeated sound events in long-duration personal audio recordings," in *Proc. IEEE ICASSP*, vol. I, 2007, pp. 233–236.

[4] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[5] M. Casey and M. Slaney, "Fast recognition of remixed music audio," in *Proc. IEEE ICASSP*, Hawai'i, 2007, pp. IV–1425–1428.

[6] R. Gribonval and S. Krstulovic, MPTK, The Matching Pursuit Toolkit, http://mptk.irisa.fr/.

[7] F. Petitcolas, MPEG for MATLAB, http://www.petitcolas.net/fabien/software/mpeg.

[8] D. Pan, "A tutorial on mpeg audio compression," *IEEE Multimedia Magazine*, vol. 2, no. 2, pp. 60–74, 1995.

[9] H. Najaf-Zadeh, R. Pichevar, L. Thibault, and H. Lahdili, "Perceptual matching pursuit for audio coding," in *Audio Engineering Society Convention, Amsterdam, The Netherlands*, 2008.