KNOWN-AUDIO DETECTION USING WAVEPRINT: SPECTROGRAM FINGERPRINTING BY WAVELET HASHING

Michele Covell & Shumeet Baluja

Google, Inc.
1600 Amphitheatre Parkway, Mountain View, CA. 94043
{covell, shumeet}@google.com

ABSTRACT

In this paper, we present a novel system for detecting known audio. We start with *Waveprint*, an audio identification system that, given a probe snippet, efficiently provides reliable forced-choice ranking of entries from an audio database. For open-set detection, we can re-examine the best-ranked matches from Waveprint using simple temporal-ordering-based processing. The resulting system has excellent detection capabilities for small snippets of audio that have been degraded in a variety of manners, including competing noise, poor recording quality, and cell-phone playback. The system is more accurate than the previous state-of-the-art system while being more efficient and flexible in memory usage and computation.

Index Terms— Acoustic Applications, Acoustic Signal Detection, Pattern Recognition, Music

1. INTRODUCTION

Detecting whether or not a snippet of audio is contained in a database of known audio is useful in many audio-identification tasks [4, 6, 10, 12], since it allows the system to move into an open-set scenario. The approach that we take is conceptually similar to one used in open-set face identification [11]. Under that approach, the system starts by forced-choice ranking of the database entries according to their probability of matching a probe. The top-N ranked matches are then used to provide score normalization. Depending on the distribution of these top scores, the probe is accepted or rejected as being in the database.

Since the forced-choice ranking provides us with the identity of our presumed song, the ordering that it provides must be extremely reliable: We cannot hope to achieve error rates that are significantly lower than those of this front-end ranking system. At the same time, since this ranking system is operating on a large database (the full population of interest), it must be computationally efficient. To allow database-size scaling, the memory used by the full database and the computation used in matching must be tightly controlled. We use *Waveprint* [1], a new approach to audio fingerprinting, since our studies have shown that it is efficient in both memory usage and computation while being highly accurate on closed-set ranking.

In this paper, we present our full detection system. We briefly review the approach to closed-set audio identification, taken by Waveprint. In Section 3, we describe the various approaches that we explored to move from the top-N (N=20) ranked list to a detection decision. In Section 4, we provide experimental results under a wide range of probe-song degradations. We include a comparison to a state-of-the-art audio-detection system [10]. We conclude with a discussion of possible future work.

2. THE WAVEPRINT SYSTEM

Since we start with forced-choice ranking and use its results to select a small subset of our known audio population for subsequent consideration, the choice of recognition system for this first stage is extremely important. Our final accuracy rates are limited by the accuracy of this initial stage. Furthermore, our final evaluation of computation and memory requirements will be at least as high as this first stage. We chose the audio-identification system, Waveprint [1], due to its efficiency and accuracy.

Waveprint uses heavily quantized (tri-level) wavelet-image decompositions of spectrogram segments to generate sparse bit vectors. Such tri-level wavelet representations have been shown to be robust in content-based image-retrieval applications [7]. Based on experimental results, we use a segment length of 1.48 seconds for this processing.

The tri-level quantizer assigns 80-98% of the wavelet coefficients to zero and the remaining coefficients to ±1. These predominantly-zero-valued coefficients are listed as sparse bit vectors, using a non-compact but simple bit-pattern encoding: each zero value is listed as "00" and ±1 are listed as "01" and "10". Following this conversion to a bit stream, only 1-10% of the data is non-zero. We use this sparseness to compact, and subsequently efficiently match, the representations using Min-Hash processing [3]. Min-Hash provides a compact fingerprint such that the similarity between two fingerprints gives a measure of the probability that the original two streams were identical. For large, sparse bit streams (such as ours), a statistically strong measure of similarity can be achieved using a comparatively compact representation: Waveprint uses p=100 independent Min-Hash counts, with each count limited to a hard maximum of 255, allowing representation in 100 bytes. We refer to these Min-Hash signatures as *sub-fingerprints*.

The final element in Waveprint is the use of location-sensitive hashing (LSH) [5] to efficiently find similar known-audio sub-fingerprints, using probe sub-fingerprints. This efficiency allows the final system to find nearest neighbors in a p=100-dimensional space by looking at (on average) only 0.26 of each million entries in the full database [1].

In summary: For robustness to noise and time or frequency distortions, Waveprint uses heavily quantized wavelet-image decompositions of spectrograms. For efficient memory usage and pairwise-efficient comparison, Waveprint then applies Min-Hash techniques to the quantized wavelet coefficients. For efficiency in retrieving candidate matches and robustness to distortions, Waveprint uses a hashing-based method for nearest-neighbor retrieval.

The above processing is applied to known-audio spectrograms with a step size of s seconds between sub-fingerprints (that is, s seconds between the starting points of neighboring, overlapping

1.48-second-long spectrogram segments). The resulting sub-fingerprints are inserted into the LSH tables for in-memory database access.

The same processing is applied to the probe-audio spectrogram. To avoid de-synchronized sampling issues, we use a smaller, *d*-ms sampling stride on the probe. Each probe sub-fingerprint is used to index into the database. The support from the resulting candidates is combined across time using both local and global dynamic time warping (DTW) constraints. Specifically, the probe-to-known mappings, formed by combining across sub-fingerprint candidate matches, are constrained to a global tempo change of less than 10% and a local match slope from probe to database position that is non-negative.

As described in [1], Waveprint has many desirable properties due to the combination of techniques that it applies. However, this combination also has a large, non-linear surface of best parameter combinations. In testing for the best tradeoff amongst forced-choice performance, memory usage, and computational load, [1] examines over 50,000 parameter combinations. The results suggest that the best operating points are around: quantization of 5% of wavelet coefficients to non-zero values; use of p=100 Min-Hash permutations to form the sub-fingerprints; subdivision of the sub-fingerprints into 25 sub-hash key inputs for LSH with a minimum of 5 sub-hash votes for each sub-fingerprint addition to the candidate match list; and on the order of 100,000 hash bins in each of these 25 LSH sub-tables. For this paper, we will report results using these parameters. Since the probe-snippet length is typically defined by the application, we consider three different lengths: 10 sec, 30 sec, and 60 sec. Since s (the database sampling) directly affects memory usage and since both s and d (the snippet sampling) affect computational load and accuracy, we also examine distinct values for these: specifically, probe sampling of d = 23.2 ms, 46.4 ms and 92.8 ms and database sampling of s = 0.46 sec and 0.92 sec. We did not examine database sampling strides longer than s = 0.92 sec since, from closed-set experiments [1], the performance of the ranking task drops by 10% (absolute error) for database sampling that is coarser than a second. This closed-set performance will directly affect our open-set performance due to its position as a first-stage process.

3. VERIFICATION PROCESSING

Once the force-choice ranking is available, we can take the top-N songs and treat the problem as a verification process. Since the number of songs that are now being considered is much smaller, we can use more elaborate processing without having a strong impact on the overall speed of the detection system: we do not have to worry about scaling with database size, since we are no longer working with the full database. Furthermore, if secondary storage is available, we can access additional data about each of the target/cohort songs with little impact on the overall system: we do not need to fit this addition information into main memory, since we can read from disk for N=20 cohort songs without significant disk-access overhead.

This freedom has led us to examine 3 different options for signal generation to be used in verification:

1. *Waveprint-only approach*: This is a baseline approach. It does no additional processing and instead uses the already-generated quality-of-match scores (and their ratios) as our input to a pass/fail classification system. This choice is clearly lowest cost in terms of computation (no additional) and disk usage (none).
2. *Original-spectrogram approach*: This approach is at the opposite extreme. It retrieves the actual target/cohort songs from disk and compares those directly with the probe snippet. This requires disk storage for all the songs in the database, since we do not know which of these will be in the target/cohort set. However, it only requires disk access and computation time for the N=20 songs that are ranked highest by Waveprint.
3. *Time-indexed Waveprint approach*: This approach falls between the two above extremes. It uses the same sub-fingerprint database as is used by Waveprint. It uses the temporal ordering of the sub-fingerprints as they were computed from original songs, along with the DTW alignment parameters, to improve our statistical support for each match score.

For both the original-spectrogram approach and the time-indexed Waveprint approach, we refined the time alignment, starting from the alignment given by the first-stage Waveprint processing. To support this, the first stage provides the location pair for the strongest sub-fingerprint match (the location within each song-probe pairing that had the smallest Hamming difference). We examine alignments that, at this pinned-location pair, are within $\pm s$ seconds of this strongest alignment (where s is the database sampling stride used in the first-stage Waveprint processing). In addition to this $2s$-offset margin, we consider matches that change the tempo by no more that 10% (that is, the wedge formed by the 90% and 110% tempo lines). In this way, using DTW, we can refine the temporal-alignment parameters from the values selected by first stage process. This refinement is called for since, for highly distorted probe snippets, the match support that is used for the original temporal alignment can be very sparse. The second stage can provide more reliable estimates, since there is more statistical support across the length of the probe snippet.

In the original-spectrogram approach, at each spectral-slice, the squared-difference term between the DTW-aligned database song and probe snippet is normalized by the inverse of the volume from the two slices being matched. These weighted mean-squared errors between the probe and database-song spectrograms are computed for each of the top N songs. Under this original-spectrogram approach to classification, we extend the vector of scores provided by Waveprint with these MSE scores. This approach is comparatively expensive, both in terms of required disk space and in terms of computation for the normalization, prior to weighted MSE comparison. However, this expense is independent of database size, making it a practical alternative.

For the time-indexed Waveprint approach, we made use of the local DTW constraint that each probe sub-fingerprint can match only one song sub-fingerprint to find the weighted least-squares-fit line to those pairs. The weighting that we use is a *scaled subfingerprint-match strength*: we reduce the strength of each of these sub-fingerprint matches according to the strength of the matches between the probe sub-fingerprint and the other (not-selected) song sub-fingerprints that were within the allowed wedge. The final output score from this stage is the mean-squared error between the match locations and the best-fit line, again weighted by the scaled subfingerprint-match strength. These output scores are used as inputs to our classification system, along with the original Waveprint scores.

Finally, we also considered two different mechanisms for categorization: (1) support-vector machines (SVMs) [2] and (2) linear regression. For the SVM classifier, we used SVM Light [8] and tried a wide variety of parameter settings both with linear and RBF kernels. For the linear-regression-based classification, we

used the classification library available in Python. For both SVM and linear regression, we provide the classifiers with a vector containing, for each of the top-20 first-stage Waveprint matches:

(*v1*) the first-stage match quality score
(*v2*) the scores in (*v1*), normalized by the best (largest) first-stage match score
(*v3*) the time extent of the first-stage match (that is, how long in seconds was the supporting DTW track, giving an indication of the matched-segment length)
(*v4*) the count of the supporting subfingerprint pairs in the first-stage match (that is, how many subfingerprints contributed to the match score)
(*v5*) if applicable, the MSE score from the second-stage processing
(*v6*) if applicable, the score in (*v5*), normalized by the best (lowest) MSE score.

This gives an 80-dimensional vector ((*v1*–*v4*) x top-20) for the Waveprint-only approach and a 120-dimensional vector ((*v1*–*v6*) x top-20) for the original-spectrogram and the time-indexed Waveprint approaches.

For all 36 cases (2 classifiers x 3 proposed second stages x 6 sampling regimes), we trained the classifier using a mixture of 1000 degraded probes, operating against a database of 35,000 minutes of music. For each probe, we ran the open-set identification twice: once with the (un-degraded) originating song in the database and once with that one song removed. To evaluate our performance, we tested on a *different* mixed set of 1000 degraded probes, again probing under both source-present and source-absent conditions.

The degradations of the probes are an equal sampling of:

(*d1*) time-offset only (part of (*d2*–*d11*) as well)
(*d2*) added echo (90% reverb after 100 ms)
(*d3*) frequency equalization (same as in [6])
(*d4*) 32-kbps MP3 (MPEG2 layer 3) transcoding
(*d5*) 4.75-kbps AMR (cell-phone) transcoding
(*d6*) loud "lyrical" added noise (from *Enya*)
(*d7*) loud "death-metal" added noise (from *Veil of Tears*)
(*d8*) -2% linear-speed modification (LSM) (slower/lower-pitch)
(*d9*) +2% LSM (faster/higher-pitch)
(*d10*) -10% time-scale modification (TSM) (slower/same-pitch)
(*d11*) +10% TSM (faster/same-pitch).

4. EXPERIMENTAL RESULTS

In our tests, there was no significant difference amongst linear-kernel SVMs, RBF-kernel SVMs, and linear regression. Given this, we use simple linear regression, as it provides fast and easily computed scores as well as high accuracy. All of the results reported in the remainder of this section use linear-regression classification.

We compared the detection performance amongst the 3 proposed second-stage approaches: Waveprint-only, time-indexed Waveprint, and original-spectrogram comparison. As expected, Waveprint-only did not match the performance of the other two approaches, having about twice the equal-error rates of the other two approaches under best conditions. Even using a 60-second probe snippet, it failed to achieve better than 98% performance, at its best equal-error-weighting setting.

More surprisingly, the time-indexed Waveprint and the original-spectrogram comparison performed at nearly identical levels. They were always within 0.5% of one another, under the equal-error-weighting criteria, with no bias on which system would do better. Given the increased cost of the original-spectrogram comparison, in both disk usage and computation, this equal performance makes the time-indexed Waveprint approach preferable. For the remainder of this section, we will consider only that option as our second stage.

We tested six combinations of database and probe sampling strides. Unlike closed-set ranking [1], the open-set performance was equal for the 3 tested probe-sampling strides, d = 92.8 ms performing as well as the others. In contrast, the second-stage open-set detection did improve with fine-grain database sampling: s=0.464 sec had an equal-error rate that was 28% lower relative to that of s=0.928 sec (1.5% lower in absolute terms). This was unexpected, since the first-stage ranking does not improve significantly with this finer-grain database sampling [1]. The improvement in classification occurred primarily due to better classification of the positive cases (when the snippets are present). The classification improvement was associated with (*v3*) for the top 5 choices. When the database sampling stride is 0.92 sec, these top matches can have a much-reduced DTW duration, due to negative interactions between this sampling and the distortions within the probe. The shorter database sampling stride (0.46 sec) provides longer temporal support to correct matches. The shorter database stride also provided a more reliable measure of the MSE distance given by the second-stage processing.

Based on these results, we selected time-indexed Waveprint, but used mixed parameters across the two stages. Under this hybrid approach, we collect database subfingerprints at the shorter stride (s=0.46 sec) and split this densely sampled database into two interleaved sets. One set is written to disk in time-sequential order. The other set is used for the first-stage processing (as well as for interleaved usage in the second stage). By saving the extra subfingerprints on disk but still having them available for temporally-ordered retrieval, we reduce the overall memory usage to 0.45 GB for a 35,000-minute database. In contrast, the fully in-memory approach would take nearly a twice as much memory (0.87 GB). The only complication to the partially in-memory approach is that we must recompute the temporal duration of the first-stage match support by extending it to include any close-enough sub-fingerprints that were in the on-disk half of the database. Since these on-disk interleaved subfingerprints are only used in the second-stage process, this approach requires only 20 seeks/reads from disk (corresponding to the top-20 ranked songs), making this approach practical even with the delays of the disk access.

Finally, we examined the relative importance of the different components in the 120-dimensional classification vector by examining the weighted variance provided to the classifier output by each dimension. To avoid drawing spurious conclusions, we retrained the linear classifier 11 times with different training sets. The majority of the classifier-output variance (78-80%) derives from (*v3*) for the top-5 ranked songs: that is, the temporal duration of the DTW tracks that support the top 5 ranked songs. The remaining classifier-output variance is almost evenly split between (*v5*), the MSE scores of the second stage, and (*v1*), the match-quality scores of the first stage. As expected, low (*v5*) MSE scores in the top-5 ranked songs support a "present" classification. More surprisingly, the classifier uses the average top-20 ranked first-stage match scores (*v1*) as evidence against a match. An intuitive explanation for this is the classifier is discounting matches of *nondescript* probes that match many songs well.

In Table 1 and Figure 1, we compare our performance to a publicly available, state-of-the-art system [10]. For these

Table 1. (% False-positive)/(% True-positive) rates for time-indexed Waveprint and for the modified Ke's system [10]

The operating point for the time-indexed Waveprint (*t-i Wave*) was the equal-error setting for the mixed-distortion probe. The operating point for Ke's system was defined by the maximum likelihood ratio from the statistical models in their codebase.

Distortion	Detection	10s probe	30s probe	60s probe
(d1) Time-offset only	*t-i Wave*	0.5/99.7	0.6/99.9	0.5/99.9
	Ke	2.9/98.7	5.9/98.7	4.6/98.8
(d2) Echo	*t-i Wave*	0.6/98.5	0.6/99.6	0.2/99.4
	Ke	2.9/97.6	5.8/98.2	4.6/98.5
(d3) Freq. Equalizer	*t-i Wave*	1.1/99.3	1.1/99.8	0.3/99.7
	Ke	2.9/97.9	6.2/98.2	4.5/98.5
(d4) 32kbps MP3	*t-i Wave*	0.4/96.9	0.4/98.5	0.4/98.7
	Ke	3.5/91.9	6.2/92.1	5.0/91.7
(d5) 4.75kbps AMR	*t-i Wave*	1.1/88.3	1.0/98.8	0.7/99.2
	Ke	2.0/77.8	4.0/94.9	3.4/96.6
(d6) Noise *(Enya)*	*t-i Wave*	0.6/73.3	0.2/94.2	0.3/96.3
	Ke	0.9/34.7	1.9/61.8	1.4/69.7
(d7) Noise *(Veil of Tears)*	*t-i Wave*	7.6/85.1	9.3/93.5	11.9/95.8
	Ke	7.3/60.7	9.7/72.0	9.2/75.4
(d8) LSM -2%	*t-i Wave*	2.4/94.2	2.6/97.0	4.4/99.0
(d9) LSM +2%	*t-i Wave*	0.7/62.2	0.3/90.0	0.4/94.3
(d10) TSM -10%	*t-i Wave*	0.6/99.4	0.7/99.4	0.6/99.4
(d11) TSM +10%	*t-i Wave*	0.6/98.0	0.8/99.6	0.6/99.6

comparisons, we used their published code base [9], with the volume normalization modified to adapt to the local 5-second RMS energy. This modification uniformly improved our results from Ke's code base [9]. Figure 1 shows the ROC curves for both Ke's and the time-indexed Waveprint systems, using 10-, 30-, and 60-sec probes under an equal-weighting mixture of distortions (*d1–d7*). This omits the time variations (LSM and TSM), since Ke's code base [9] did not include the extension needed to support timing changes. Table 1 lists performance numbers for all 11 distortion types separately. As mentioned above, these results were gathered using a test set of 2000 samples (1000 probes x present/absent). The probes were from a separate set than the training data used for the linear classifier stage.

Our detection results are consistently better than the previous system [10] on the same probe sets, showing more than 80% relative reduction in error for 30- and 60-sec probes. We perform better with a 10-sec probe than Ke's system [10] does with a 60-sec probe. Finally, we did nearly as well with 30-sec probes as with 60-sec probes, making longer probe lengths unnecessary.

In addition to accurate performance, detection systems must be able to handle large databases efficiently. Using our selected parameters, the time-indexed Waveprint system uses 0.45 GB of memory for a 35,000-minute database (about ½ for the sub-fingerprints and ½ for the LSH tables which point to them). The processing time is dominated by the first-stage Waveprint ranking system. On a single-CPU 3.4-GHz Pentium, operating against a 10,000-song database, this system operates about 14× faster than the probe-length time (that is, 46-285× faster than the average 3.5-minute song length, depending on probe length). For the selected parameter set, this speed will be largely unaffected by database size. In our tests, these memory and computation-time requirements were significantly lower than that of [10]. This claim

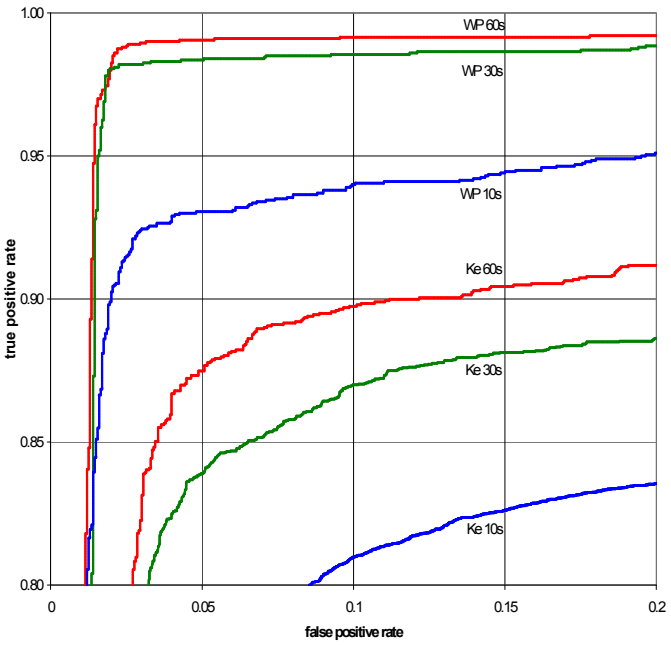

Figure 1. ROC curves for time-indexed Waveprint and for the modified Ke's system [10] on a time-distortion-free set (*d1-d7*)

is supported by memory-usage and complexity analyses of the two systems [1].

5. CONCLUSIONS

In this work, we have presented a new approach to detection of known audio. We use Waveprint ranking [1] to move the problem from detection to verification. By re-using the same sub-fingerprints that are used by the first-stage ranking system and by storing interleaved fingerprints to disk, we do not increase our memory requirements above that system's usage. Since the interleaved fingerprints are only needed on a small set (20 songs), using disk storage does not substantially affect our speed. Our only additional processing is similar to the second-look stage from [6], with the significant difference that our second-look processing is only on the top-*N* ranked songs, not the full candidate list that share sub-fingerprints with the probe.

REFERENCES

[1] Baluja, Covell. Content fingerprinting using wavelets, *CVMP* (2006).
[2] Cortes, Vapnik. Support vector networks, in *Machine Learning* (1995).
[3] Cohen, *et al.* Finding interesting associations without support pruning. *Knowledge & Data Engineering,* vol 13, no. 1 (2001).
[4] Fink, *et al.* Social- and interactive-television applications based on real-time ambient-audio identification, *EuroITV* (2006).
[5] Gionis, *et al.* Similarity search in high dimensions, *VLDB* (1999).
[6] Haitsma, Kalker. Highly robust audio fingerprinting. *ISMIR* (2002).
[7] Jacobs, *et al.* Fast multiresolution image querying, *SIGGRAPH* (1995)
[8] Joachims. Making large-scale SVM learning practical. in *Advances in Kernel Methods – Support Vector Learning* (1999)
[9] Ke, *et al.* Computer vision for music identification: server code, http://www.cs.cmu.edu/~yke/musicretrieval/musicretr-1.0.tar.gz (2005).
[10] Ke, *et al,* Computer vision for music identification. *CVPR* (2005).
[11] Li, Wechsler. Open-set face recognition using transduction. *PAMI* vol. 27, no. 11 (2005).
[12] Shazam Entertainment Ltd. http://www.shazam.com/ (2006).