MASK: Robust Local Features for Audio Fingerprinting

Xavier Anguera, Antonio Garzon[†] and Tomasz Adamek[‡]
Telefonica Research,
Torre Telefonica Diagonal 00,
08019, Barcelona, Spain
xanguera@tid.es

***Abstract*—This paper presents a novel local audio fingerprint called MASK (Masked Audio Spectral Keypoints) that can effectively encode the acoustic information existent in audio documents and discriminate between transformed versions of the same acoustic documents and other unrelated documents. The fingerprint has been designed to be resilient to strong transformations of the original signal and to be usable for generic audio, including music and speech. Its main characteristics are its locality, binary encoding, robustness and compactness. The proposed audio fingerprint encodes the local spectral energies around salient points selected among the main spectral peaks in a given signal. Such encoding is done by centering on each point a carefully designed mask defining regions of the spectrogram whose average energies are compared with each other. From each comparison we obtain a single bit depending on which region has more energy, and group all bits into a final binary fingerprint. In addition, the fingerprint also stores the frequency of each peak, quantized using a Mel filterbank. The length of the fingerprint is solely defined by the number of compared regions being used, and can be adapted to the requirements of any particular application. In addition, the number of salient points encoded per second can be also easily modified. In the experimental section we show the suitability of such fingerprint to find matching segments by using the NIST-TRECVID benchmarking evaluation datasets by comparing it with a well known fingerprint, obtaining up to 26% relative improvement in NDCR score.**

***Keywords*-Audio fingerprinting, audio indexing, copy detection**

I. Introduction

In this paper we propose a novel audio fingerprint we call MASK (which stands for Masked Audio Spectral Keypoints). Audio fingerprinting is understood as the method by which we can compactly represent an audio signal so that it is convenient for storage, indexing and comparison between audio documents. It differs from watermarking techniques [1] in that no external information/watermark needs to be a priori encoded into the audio, as the audio itself acts as the watermark.

A good fingerprint should capture and characterize the essence of the audio content. More specifically, the quality of a fingerprint can be measured in four main dimensions: discriminability, robustness, compactness and efficiency. A fingerprint has a high discriminatory power if two fingerprints extracted from the same location in two audio segments coming from the same source are very similar, and at the same time, fingerprints extracted from segments coming from different sources, or different locations in the same source, are very different. Another important quality is robustness to acoustic transformations. We define as a transformation any alteration of the original signal that modifies the physical characteristics of the signal but still allows a human to judge that such audio comes from the original signal. Typical transformations include MP3 encoding, sound equalization and mixing with external noises or signals. Next, compactness is also important for reducing the amount of information that needs to be compared when using fingerprints for searching in large collections of audio documents. Finally, efficiency refers to how fast the fingerprint can be extracted from the original signal and, equivalently, the efficiency of retrieval methods that can be used with such fingerprint.

In recent years there have been many proposals for different ways to construct acoustic fingerprints [2]–[6]. For an early review of some alternatives see [7]. Some of them are not robust enough to severe audio transformations, their performance degrades when encoding content other than music or are expensive to compute or to store. Three of the most cited audio fingerprints are probably the Shazam fingerprint presented in [2], the system proposed by Philips in [3] and the waveprint, proposed by Google [4].

The Shazam fingerprint [2] encodes the relationship between two spectral maxima, where one of them is called an anchor point. By encoding multiple maxima in a single fingerprint they are prone to errors when either of the maxima is missing. For this reason, in order to make the system robust, for each selected anchor point they need to store several tuple combinations within its target area, creating an overhead of data to be stored for a given audio. In addition, they encode the data inside the fingerprint in 3 different blocks (20 bits for the frequency locations of the two peaks and 12 bits for their time difference). If the comparison between fingerprints is allowed some error they need to first apply a conversion from binary form to the corresponding natural numbers and later differentiation to find how far the spectral maxima are from each other. Given

[†]For the duration of this project Antonio Garzon was a visiting scholar from Universitat Pompeu Fabra

[‡]Tomasz Adamek is currently with www.catchoom.com

that the fingerprint comparison step is the most repeated step in any retrieval algorithm it would be much better if such comparisons could be performed entirely in the binary domain or lead by simple comparison table lookups (which is unfeasible here due to the big number of bits used in the frequency and time encoding).

The Philips fingerprint [3] encodes the signal sequentially using a fixed step size, which reduces its flexibility to adapt its storage requirements to different application scenarios while still obtaining a compatible fingerprint. For example, for a server-based solution without any storage problems it is desirable to store as many fingerprints as available, while for a solution embedded in a mobile device it would be better to reduce the number of computed fingerprints to save on computation and bandwidth when sending them to a server for comparison with a database. In the Philips system one can only achieve this by modifying the fingerprint extraction step size, although this can severely change the resulting fingerprints and thus the final performance. Furthermore, in the encoding step, Philips solution relies on the energy differences between pairs of band energies, and encodes all bands in each time step. Taking hard binary decisions in the comparison of the values of two adjacent bands is prone to errors due to small fluctuation in the signal. This can cause instability in certain bits and affect its robustness. In addition, by encoding all the bands in the spectral domain at every analysis step the system is more prone to errors in regions where the SNR is low and where differences in energy are due to very small energy noises added to the signal, which change arbitrarily depending on the transformations applied to the audio.

Finally, the Google waveprint fingerprint [4] proposes an alternative encoding of the audio by using the wavelet transformation. Such approach is indirectly encoding the peaks in the spectra as indicated by the biggest coefficients in the wavelet domain. Even though their approach seems more robust than the previous two approaches, it is computationally very expensive and results in a high number of bits per fingerprint, thus making its computation in an embedded platform or its transmission through slow channels (for example the mobile network) very impractical.

The proposed MASK fingerprint tries to address the shortcomings of the previously cited algorithms. Like in most cases, we also rely on the fact that spectral maxima are usually resilient to transformations applied to the audio. In MASK we first select salient point chosen from the maxima in the Mel-filtered spectral representation of the signal. By modifying parameters used in the salient point selection criteria we can adapt the system to any given application requirements (obtaining more or less density of keypoints per second). Next, we encode the region around each selected point by superimposing a mask centered on it and comparing the energy of selected mask regions, as defined by the mask designer. By encoding each salient point (and its surroundings) independently we are more localized that the Shazam fingerprint and we believe we require less fingerprint vectors to encode the audio with similar robustness. The final fingerprint encodes the binary comparison of region energy differences together with the Mel band where the peak is found. Given that the number of Mel filters used is small (usually 16 or 32) it is later very fast to compute the difference between two fingerprints using a lookup table.

The rest of the paper is structured as follows: Section II explains the process used to obtain the MASK fingerprint from the audio data. Section III described how to index and search for matching sequences of audio by using the proposed fingerprint. Then, in Section IV we explain the experiments we performed to test the fingerprint and comparisons with the Philips fingerprint, and we finally draw some conclusions and point some possible future directions.

II. MASK FINGERPRINT

In the following sections we describe in detail the extraction of the proposed MASK fingerprint from an audio signal. The processed signal can either be a static file (where we know a priori its start and end times) or streaming audio as the fingerprint is extracted only considering local information.

As seen in figure 1 the MASK fingerprint extraction method is composed of 4 main blocks. First, the input signal is transformed from the time domain to the spectral domain, and transformed into Mel-scale. Then, spectral salient points are selected. Next, for each one of the salient points we apply a mask around it and perform the grouping of the different spectrogram values into regions, as defined by such mask. Finally, we compare the averaged energy values of each one of these spectrogram regions and construct a fixed length binary descriptor. This local descriptor forms the proposed MASK fingerprint. Next sections describe in more detail each one of these steps.

A. Time-to-Frequency Transformation

Given any input acoustic signal, we first down-sample it to 4KHz and band-pass filter it to focus only on the range between 0.3KHz and 3KHz, similarly to what is done in the Philips fingerprint proposed in [3]. Then, in order to find the spectral peaks we compute the short term-FFT (Fast Fourier Transform) on the signal at fixed time intervals (10ms) and using a short-term window (100ms). Such parameters were defined experimentally and similarly to the values used in [2]–[4]. Note that before we apply FFT we filter every block by a Hamming filter in order to reduce discontinuities at the edges. Then we apply a Human Auditory System (HAS) filtering to equalize the frequency bins to values that correspond to the human perception of audio, and to reduce the number of total frequency bins. From all possible alternatives, in this work we use the MEL filter-bank with

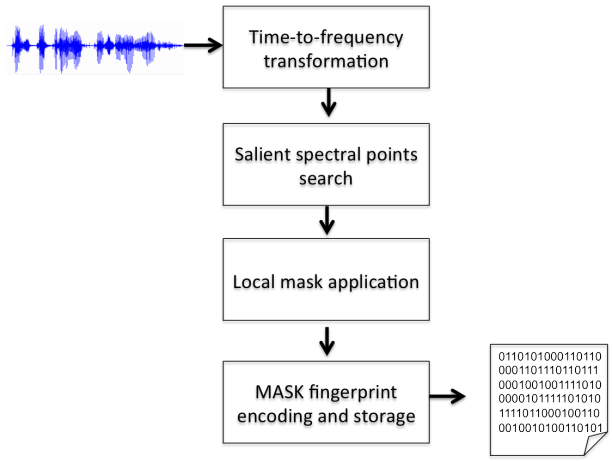


Figure 1. *Block diagram of the steps involved in the MASK fingerprint extraction.*

18 bands. Note that bigger bandwidths and a higher number of bands could be used without any necessary other change to the algorithm.

B. Selection of Salient Spectral Points

Once the spectral representation of the signal has been obtained, we need to select salient points in the spectral domain where to center the computation of the proposed MASK fingerprint. In our implementation we focus on the local maxima which, like [2]–[4], we found to be resilient to many audio transformations. In general, a local spectral maxima or spectral peak can be defined as any point in frequency whose energy is greater than the points adjacent to it, either in frequency, time or in both.

In our current implementation the peaks selection method is quite simple. A time-frequency position in the spectrogram is selected as a peak if its energy is strictly bigger than all energies in immediately adjacent band-time locations. Note that we never place a peak in the top or bottom-most MEL bands, leaving only 16 possible frequency peak positions when using 18 Mel bands (aside from slightly reducing the bandwidth being used to obtain the information from the file, this does not have any implications on the way information is obtained from the audio signal). Our observations indicate that a good coverage of the audio is obtained by extracting between 70 to 100 peaks per second. In order to limit the number of peaks selected as salient points we apply a post-detection filtering to select only those peaks whose energy stays above a given temporal masking threshold, defined according to its distance to the previously selected peak in the same MEL band. Equation 1 shows the threshold we use in our implementation.

$$Thr[n] = \alpha^{\Delta t} E[n-1] \exp - \frac{(\Delta t)^2}{2 * \sigma^2} \qquad (1)$$

where Δt is the distance (in frames) between the previous peak and the considered peak, $E[n-1]$ is the energy of the previously selected peak and α, σ are two parameters used to set the threshold falling rate and its width, respectively. In the proposed implementation we set them to 0.98 and 40 following a similar implementation used in the Shazam fingerpring by Dan Ellis[1].

C. Spectrogram Masking Around Salient Points

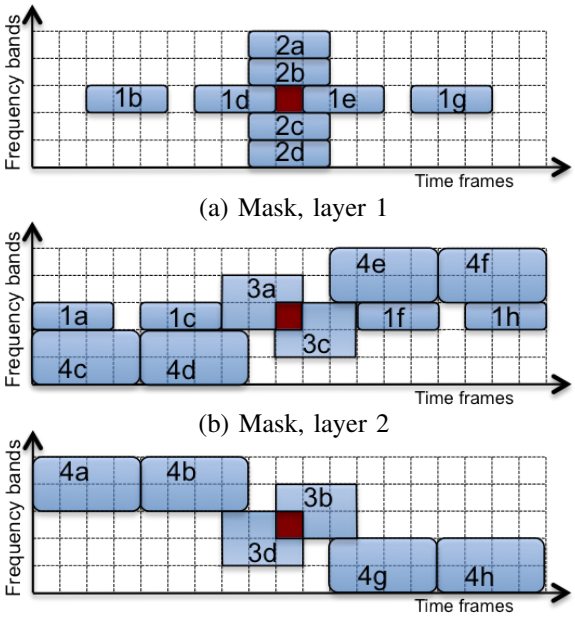
Once the salient points have been selected we apply a mask centered at each of them. This defines regions of interest around each point that are used for encoding the resulting binary fingerprint. The encoding is carried out by comparing differences in average energies between certain defined region pairs. A region in the mask is defined as either a single time-frequency value or a set of values that are considered to contain similar characteristics (they are usually contiguous in time and/or frequency). When a region is composed of several values its energy is represented by its arithmetic average. Note that the regions defined by the mask are allowed to overlap with each other. The optimum location and size of each region in the mask, as well as the total number of regions, can vary depending on the kind of audio that is being analyzed and the number of total bits we desire for the fingerprint. How to automatically determine these remains as future work at the time of writing this paper. The particular mask we used is shown in Figure 2 (split into three diagrams for a better visualization). This mask covers 5 MEL frequency bands around the peak (2 bands above and 2 bands below) and extends for 190ms (90ms. before and 90ms. after). Different regions grouping together several spectral values are labeled using a numeric value followed by a letter, used in the next section.

Note that when a salient peak is found either at band N-1 or at band 2 (i.e. with only one band above or below it) the mask in Figure 2 will either have the first or last rows falling outside of the spectrogram limits. In such case we duplicate the values of the first/last available bands to cover the inexistent values for the first/last mask rows. The way we define the different regions allows for these cases not to affect much the properties of the resulting fingerprints. Note also that although the energy points in the first and last band of the MEL spectrogram are not allowed as salient points, they are used in the fingerprint encoding.

D. Fingerprint Construction

In this step we construct the fingerprint as follows: first, a block of 4 bits is inserted encoding the location of the salient peak within the 16 MEL-filtered spectral bands where maxima can be located (excluding the first and last computed bands). Next, we insert the binary values resulting from the comparison of selected regions around the salient peak, as

[1]http://labrosa.ee.columbia.edu/matlab/fingerprint/



(a) Mask, layer 1

(b) Mask, layer 2

(c) Mask, layer 3

Figure 2. Frequency mask covering a region of 5 bands per 19 temporal frames, split into 3 layers to better observe the overlapping regions

defined by the mask. Table I shows the region comparisons we used in this paper, where we split the obtained bits into 5 different groups. The first and second groups encode the horizontal and vertical evolution of the energy around the salient peak. The third group compares the energy around the most immediate region around the salient peak, while the fourth and fifth groups encode how the energy is distributed along the furthest corners in the mask. In total, in Table I we define 22 bits. More bits can be easily obtained by defining other region comparisons.

Note that in comparison to fingerprints obtained by encoding the audio a regular intervals (e.g the Philips fingerprint) we only encode those locations with certain spectral properties, defined by the spectral energy value and the proximity to prior maxima in the same band. As a result we are flexible in defining how many fingerprints we desire to obtain per second and, by defining a proper mask, how accurately we want to encode its surroundings.

III. Fingerprint indexing and comparison

The proposed fingerprint allows for using indexing techniques common when using local features (e.g. [2]). For every extracted fingerprint we can index it in a hash table as the hash key. The corresponding hash value is then composed of two terms: (i) the ID of the audio material the fingerprint belongs to, and (ii) the time elapsed from the beginning of the audio material in which the salient peak has been found. Retrieval of acoustic copies can be implemented in a standard way by defining an appropriate distance between any pair of two fingerprints.

The particular matching algorithm we used is very similar to the one we used in [8] on features similar to [3]. The algorithm is composed of two steps, in a first step we retrieve all the exact matches to every MASK fingerprint in a given query. The time-difference between every matching query-reference fingerprint is then used to find those segments in the query that are aligned to segments in the reference collection. We allow any matching segment to have up to 5 seconds distance between two consecutive matching keypoints (if the distance is higher they are treated as two different matches). In a second step, we retrieve the segment with most matching fingerprints and it is further processed to obtain a more accurate similarity score. For each fingerprint in the selected query segment we compute the hamming distance with the corresponding fingerprint in the reference segment. Those fingerprints with hamming distance smaller than 4 (up to 3 erroneous bits) are encoded as 1, and the rest as 0, in a temporary binary vector. The final score is the average value over this binary vector for a window of length 5 seconds. This returns a score bounded from 0 to 1.

IV. Experimental section

In order to evaluate the suitability of the proposed MASK fingerprint we performed tests using the NIST-TRECVID benchmark evaluation [9] datasets from years 2010 and 2011. The task in these evaluations is, given a query video, to detect whether any segment within appears in the reference database. In order to make the task more realistic the query videos have been tampered with several audio and video transformations. In the audio domain there are 7 transformations possible, which are described in the Trecvid website[2]. The reference database contain videos downloaded from an online internet archive, simulating the general video content that can be found online in websites like YouTube. The reference database is composed of over 400 hours of videos. Although for every evaluation year there are around 11K queries, in our experiments we only used the ones with unique audio content, which are around 1.4K per year.

To evaluate the performance of the system we use the The Normalized Detection Cost Rate (NDCR). The NDCR score is the main performance metric used in the TRECVID evaluations and is defined as a weighted cost function of the probability to miss the detection of an existing copy (P_{miss}) versus the probability to falsely indicate that there is a copy in the database for a given query (P_{FA}). The lower the NDCR value, the more effective the detection is. In the Trecvid evaluations two different profiles of weights are defined for the P_{miss} and P_{FA}. In our experiments we considered only the balanced profile, where both probabilities are considered more equally (although P_{FA} still plays an important role).

[2]www-nlpir.nist.gov/projects/tv2008/pastdata/copy.detection/gt/a.gt.details

Table I
REGION PAIRS COMPARISONS FOR THE EXAMPLE MASK

Bit number	type	Comparisons
1 to 7	Horizontal max	1a – 1b, 1b – 1c, 1c – 1d, 1d – 1e, 1e vs, 1f, 1f – 1g, 1g – 1h
8 to 10	Vertical max	2a – 2b, 2b – 2c, 2c – 2d
11 to 14	Immediate quadrants	3a – 3b, 3d – 3c, 3a – 3d, 3b – 3c
15 to 18	Extended quadrants 1	4a – 4b, 4c – 4d, 4e – 4f, 4g – 4h
19 to 22	Extended quadrants 2	4a+4b – 4c+4d, 4e+4f – 4g+4h, 4c+4d – 4e+4f, 4a+4b – 4g+4h

In order to compare results with the state of the art we implemented the Fingerprint proposed in [3], which we refer to as the Philips fingerprint. In our implementation we computed 16 Mel Frequency bands over 25ms hamming-filtered audio segments, obtained every 10ms. Then we computed a 15-bit binary fingerprint by comparing the energies in adjacent bands in the way proposed by the authors. In order to make the MASK fingerprint as similar as possible to the baseline, we also extracted 18 Mel bands from which we obtained the spectral maxima and obtained a 22bit fingerprint from each one. Although the number of bits per fingerprint is bigger in MASK, the density of fingerprints per second tends to be smaller for MASK (80 to 100 versus 100 in Philips) which results in feature files of similar length for each case. The indexing and search of Philips fingerprints is implemented as described in [8], which is very similar to the MASK implementation.

A. Experimental results

Tables II and III present the results for the proposed fingerprint and the Philips fingerprint considering both the minimum NDCR score (in II) and the actual NDCR (in III). We include both tables because although the (official) minimum NDCR can be used as a good metric of performance, it is automatically computed by setting a different optimum threshold for each transformation, which is not realistic in a real-life scenario. The actual NDCR table was computed by manually setting the same threshold for all transformations, set to the average of the optimum threshold in each transformation. In addition, both tables report on results obtained when returning only the best match for every query and when returning the best 20 matches. In the Trecvid evaluation at maximum one copy is expected, thus returning only the 1-best usually result in better scores. Returning 20-best results makes more sense when performing a search task, where more than one match is likely to occur. Note that NDCR results deteriorate greatly between 1-best and 20-best cases due to the high impact that false alarms have in the NDCR metric, compared with the low impact of missing a true copy.

Exploring the results in more detail, generally in all cases the MASK fingerprint outperforms our implementation of the Philips fingerprint. The relative percentage improvement is shown in the last column of both tables, and are higher for the 2010 dataset than for the 2011 dataset. We found no explanation for this, as NIST claims that both sets of queries are created in a similar manner. The best results using the MASK fingerprint are around 0.4 which is close to state-of-the-art performance on the TRECVID datasets, using only the audio modality (results using multimodal inputs improve quite a bit). Comparing the results from both tables we see that the 1-best results for MASK are pretty stable, loosing 0.02 points in NDCR in the worst case. This is explained by the low standard deviation values of the optimum thresholds for each modality shown in the 4th column in table III, as compared to the much higher values in the Philips descriptor.

Next, Figure 3 shows the Min NDCR scores for all 7 audio transformations on the 2010 and 2011 TRECVID datasets for the MASK and Philips fingerprints. As expected, transformation 1 is the best in all cases (the query has not been altered from the reference) and the last 3 transformations (containing external overlapped speech) generally achieve the worst results. In general, the MASK fingerprint obtains better results than the Philips fingerprint, except for transformation 6 (mix with speech and multi-band compression), which obtains worse results in 2010 and similar in 2011. We are still to understand why MASK obtains worse results in this transformation.

Table II
COMPARISON OF MINIMUM NDCR SCORES

system	# results	dataset	Min. NDCR	% improve.
Philips	1	2010	0.55	–
MASK			**0.43**	21.8%
Philips	20	2010	1.03	–
MASK			0.79	23.3%
Philips	1	2011	0.53	–
MASK			**0.48**	9.4%
Philips	20	2011	0.96	–
MASK			0.82	14.5%

Table III
COMPARISON OF ACTUAL NDCR SCORES

system	# results	dataset	Thr. std.	Act. NDCR	% improve.
Philips	1	2010	0.11	0.60	–
MASK			0.03	**0.44**	26.6%
Philips	20	2010	0.12	1.19	–
MASK			0.04	0.91	23.5%
Philips	1	2011	0.08	0.57	–
MASK			0.03	**0.50**	12.2%
Philips	20	2011	0.09	1.18	–
MASK			0.06	1.02	13.5%

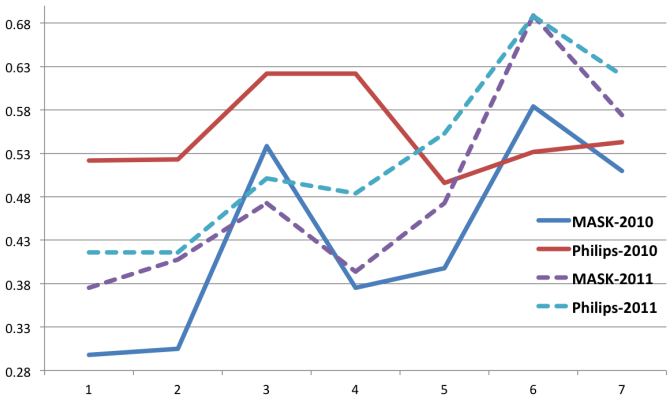


Figure 3. *Min NDCR score per transformation.*

Finally, Figure 4 shows the score histograms for the best matching reference segments given all queries, for the MASK and Philips fingerprints, as computed on the TRECVID 2011 dataset. All scores are bounded from 0 to 1, being those close to 1 indicating better matches. According to the ground truth approximately 70% of the queries contain a match in the reference dataset. We can see how the Philips fingerprint shows a much less discriminative histogram as it is relatively flat throughout the whole range of scores. Alternatively, the MASK fingerprint shows a clear bimodal distribution, which can be attributed to queries with a clear match versus queries with no clear match. The number of queries with no-clear matches is currently around 50%, which indicates that some queries can not find the real match and return some low-score alternative. Finally, note that the optimum average score used in Table III is 0.27 for MASK, which coincides with the place where the low-scored mode meets the high scores in Fig. 4.

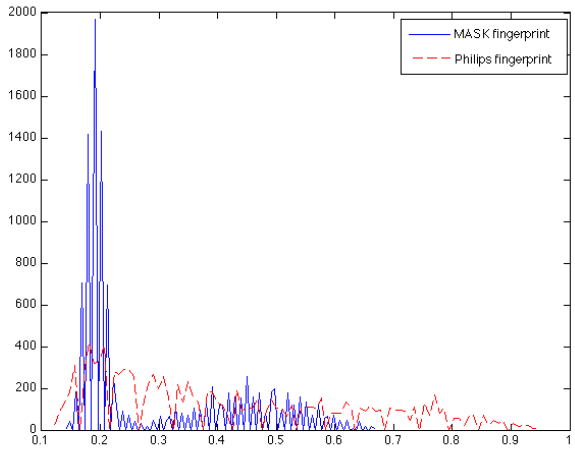


Figure 4. *Scores histograms for the 1-best results on MASK and Philips fingerprints.*

V. Conclusions

In this paper we have presented a novel local audio fingerprint called MASK (Masked Audio Spectral keypoints) that is able to encode with few bits the audio information of any kind in an audio document. The fingerprint is designed to address the problems we observed in popular audio fingerprints we reviewed. In particular, we focused on locality, binary encoding, robustness and compactness. MASK fingerprints encode the local energy distribution around salient spectral points by using a compact binary vector. Given the salient point selection process, we can easily adapt the number of descriptors per second to match any particular application need. On each salient point, a mask is used to define energy regions whose energy is compared and encoded using one bit each. The final fingerprint encodes these comparisons together with the frequency band where the peak was found. Such fingerprint is resilient to several transformations of the original audio and works on all sorts of audio, including speech, music and general sounds. We tested the fingerprint using TRECVID 2010 and 2011 video-copy detection datasets, comparing our proposal with the approach shown in [3], obtaining consistently better results.

References

[1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings IEEE, Special Issue on Identification and Protection of Multimedia Information*, vol. 87, no. 7, pp. 1079–1107, 1999.

[2] A. Wang, "An industrial strength audio search algorithm," in *Proc. ISMIR, Baltimore, USA*, 2003.

[3] J. Haitsma and A. Kalker, "A highly robust audio fingerprinting system," in *Proc. International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[4] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision and data-stream processing," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[5] V. Gupta, G. Boulianne, and P. Cardinal, "Content-based audio copy detection using nearest-neighbor mapping," in *ICASSP*, 2010, pp. 261–264.

[6] A. Saracoglu, E. Esen, T. Ates, B. Acar, U. Zubari, E. Ozan, E. Ozalp, A. Alatan, and T. Ciloglu, "Content based copy detection with coarse audio-visual fingerprints," in *Proc. CBMI*, 2009, pp. 213–218.

[7] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in *Proc. International Workshop on Multimedia Signal Processing*, 2002.

[8] E. Younessian, X. Anguera, T. Adamek, N. Oliver, and D. Marimon, "Telefonica research at trecvid 2010 content-based copy detection," in *Proc. NIST-TRECVID Workshop*, 2010.

[9] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. New York, NY, USA: ACM Press, 2006, pp. 321–330.