Lossy Audio Compression Identification

Bongjun Kim
Northwestern University
Evanston, USA
bongjun@u.northwestern.edu

Zafar Rafii
Gracenote
Emeryville, USA
zafar.rafii@nielsen.com

***Abstract*—We propose a system which can estimate from an audio recording that has previously undergone lossy compression the parameters used for the encoding, and therefore identify the corresponding lossy coding format. The system analyzes the audio signal and searches for the compression parameters and framing conditions which match those used for the encoding. In particular, we propose a new metric for measuring traces of compression which is robust to variations in the audio content and a new method for combining the estimates from multiple audio blocks which can refine the results. We evaluated this system with audio excerpts from songs and movies, compressed into various coding formats, using different bit rates, and captured digitally as well as through analog transfer. Results showed that our system can identify the correct format in almost all cases, even at high bit rates and with distorted audio, with an overall accuracy of 0.96.**

***Index Terms*—lossy compression, audio coding format.**

I. Introduction

The objective of data compression is to reduce the size of data for more efficient storage or transmission. While lossless compression encodes data in a reversible manner, lossy compression removes perceptually less significant information, typically to further reduce the size. In audio, lossy coding formats are widely used, in music and video files or radio and television broadcasting, with the popular ones being MP3, AAC, AC-3, Vorbis, and WMA. Compression algorithms which can encode to such formats first transform the audio signal into a time-frequency representation, derive a psychoacoustic model to locate regions of perceptually less significance, then quantize the data given that psychoacoustic model, and finally convert it into a bitstream [1].

Audio compression identification is thus defined as the identification of information regarding the data compression that an audio signal has undergone, regardless of the content. This can be, for example, the bit rate at which the audio data was encoded, the parameters used at the time-frequency analysis stage, or even the exact samples in the audio signal where the framing took place before the windowing and transform were applied. Applications include detection of alterations in the audio data, authentication of the audio quality, or even reverse-engineering of the encoding process.

The first ideas for identifying the framing from an audio signal that has undergone lossy compression, in the case of the MPEG-1 filter bank, were proposed in [2]. The goal was to limit the accumulation of distortion when repeating encoding and decoding. The first algorithm for identifying the compression parameters from an audio signal, based on AAC, was presented in [3]. The first implementation of that work, based on MP3, was then proposed in [4]. The idea was to search for the compression parameters and framing conditions which match those used for the encoding, by measuring traces of compression in the audio signal, which typically correspond to time-frequency coefficients quantized to zero.

The first work to investigate alterations, such as deletion, insertion, or substitution, in audio signals which have undergone lossy compression, namely MP3, was presented in [5]. The idea was to measure traces of compression in the signal along time and detect discontinuities in the estimated framing. An extension of this work, using both MP3 and AAC, was later proposed in [6]. The first work to detect audio files which have been re-encoded using lossy compression, again MP3, from a lower bit rate to a higher bit rate was then presented in [7]. The idea here was to use statistical features derived from the time-frequency analysis of the audio signal to distinguish between single and double-compressed files. Similar works for identifying multiple compressions, in the case of MP3, were also proposed in [8], [9], [10], and [11], with most of them using support vector machines (SVM) for classification, and including the detection of the original bit rates.

Other works have also been proposed for various audio compression identification tasks. In [12], a method was proposed for identifying the coding formats and bit rates from the bit stream, using statistical features and an SVM, and tested with various coding formats, including MP3, AAC, Vorbis, and WMA, for both single and double-compressed data. In [13], that work was extended with another method using audio quality measures. In [14], approaches were presented for detecting multiple compressions, in the case of MP3, by using statistical features and various machine learning algorithms. In [15], a method was proposed for identifying the bit rates in the case of MP3, AAC, and WMA, as well as the coding format, by also using statistical features and an SVM. In [16], the first method based on a convolutional neural network (CNN) was proposed for estimating the bit rates, in the case of AAC. In [17], a method was proposed, also based on a CNN, for detecting if lossy compression was used at all, for various coding formats, including MP3, AAC, AC-3, Vorbis, and WMA, and different bit rates, including high bit rates where traces of compression are scarce.

Based on the ideas presented in [3], [4], we propose to analyze an audio recording which has undergone lossy compression by searching for the compression parameters and framing conditions which match those used for the encoding. More specifically, the idea is to analyze the audio signal following the time-frequency analysis of a common lossy audio compression algorithm, for given sets of parameters associated with known coding formats, and measure traces of compression for different framing conditions. Because traces of compression become visible only when the compression parameters and framing conditions match those used for the encoding, we are then able to infer them and identify the corresponding lossy audio coding format.

Besides from detecting the correct compression parameters and coding formats, this system can also find application in identifying the original source of an audio signal. For example, provided that different TV channels and streaming services use different compression parameters and coding formats when encoding their audio content, we could identify the source of a signal by analyzing traces of compression within it and infer the correct channel or service, even if the same content can be played in different TV channels and streaming services. Such a system can therefore be particularly helpful for tasks such as media monitoring and audience measurement.

The contributions of our approach are the following. First, we propose a new metric for measuring traces of compression which is robust to variations in the audio content. We also propose a new method for combining the estimates from multiple audio blocks which can refine the results. Then, we evaluated our system for the five popular lossy audio coding formats, namely MP3, AAC, AC-3, Vorbis, and WMA, and with different bit rates, including high bit rates for which traces of compression will be scarce. Finally, we used audio recordings captured digitally, as well as through analog transfer which will introduce sample desynchronization and electronic noise in the audio. In a real-world scenario, digital and analog transfer could correspond to audio captured through HDMI and analog output, respectively, for example, on a set-top box. As far as we known, the task of distinguishing between all the popular lossy audio coding formats, including high bit rates and distorting analog transfer, has never been done before.

The rest of the article is organized as follow. In Section II, we describe lossy audio compression and the popular lossy audio coding formats. In Section III, we present our lossy audio compression identification system, including the post-processing stage. In Section IV, we propose an evaluation of our system for identifying lossy audio coding formats, with a comprehensive dataset. In Section V, we conclude this article.

II. Lossy audio compression

Lossy audio coding formats are widely used for storage, e.g., in music and video files, and transmission, e.g., in radio and television broadcasting. MP3, AAC, AC-3, Vorbis, and WMA are certainly the most popular of those formats. Compression algorithms which can encode to such formats first transform the audio signal into a time-frequency representation, typically by using a pseudo quadrature mirror filter (PQMF) bank, the modified discrete cosine transform (MDCT), or a hybrid filter bank which combines PQMF and MDCT (e.g., in MP3) [1]. Equation (1) shows the computation of the MDCT.

$$\underset{0\leq k<\frac{N}{2}}{X_k} = \sum_{n=0}^{N-1} x_n cos\left[\frac{2\pi}{N}\left(n+\frac{1}{2}+\frac{N}{4}\right)\left(k+\frac{1}{2}\right)\right] \tag{1}$$

A psychoacoustic model is derived in parallel to the time-frequency decomposition to locate regions of perceptually less significance, which will be ignored or quantized using fewer bits. This will lead to traces of compression in the audio signal, which can be visible in the derived time-frequency representation, typically as high-frequency cuts, ruptures between frequency bands, and time-frequency holes. The data is then quantized according to the psychoacoustic model and finally converted into a bitstream. Additional techniques can also be used to improve the encoding, e.g., joint stereo coding, temporal noise shaping, spectral band replication, etc. [1].

Different coding formats will typically depend on different parameters to encode the audio data. In particular, different window lengths and window functions will be used at the time-frequency analysis stage. In some cases, those parameters can also adapt to the audio content; shorter windows can be used for transients, and hybrid windows between short and long windows. Window functions such as the sine, slope, and Kaiser-Bessel-derived (KBD) windows are typically used by the popular lossy audio coding formats, i.e., MP3, AAC, AC-3, Vorbis, and WMA [1]. Equations (2), (3), and (4) show the computation of the sine, slope, and KBD windows, respectively.

$$w_n = \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right] \quad 0\leq n<N \tag{2}$$

$$w_n = \sin\left(\frac{\pi}{2}\sin^2\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right]\right) \quad 0\leq n<N \tag{3}$$

$$\begin{gathered}
w_n = \begin{cases} \sqrt{\frac{\sum_{i=0}^{n} w'_i}{\sum_{i=0}^{\frac{N}{2}} w'_i}} & 0\leq n<\frac{N}{2} \\ \sqrt{\frac{\sum_{i=0}^{N-1-n} w'_i}{\sum_{i=0}^{\frac{N}{2}} w'_i}} & \frac{N}{2}\leq n<N \\ 0 & \text{otherwise} \end{cases} \\
w'_n = \begin{cases} \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2\pi}{N-1}-1\right)^2}\right)}{I_0(\pi\alpha)} & 0\leq n<N \\ 0 & \text{otherwise} \end{cases}
\end{gathered} \tag{4}$$

I_0: 0^{th}-order modified Bessel function of the first kind

More specifically, MP3 uses 1152 samples (384 samples for the short window) and a sine window [1], [18], [19]. AAC uses 2048 samples and a sine or KBD window with α = 4 (256 samples and KBD window with α = 6 for the short window) [1], [19], [20]. AC-3 uses 512 samples (256 samples for the short window) and a KBD window with α = 5 [1], [21], [22]. Vorbis uses between 64 and 8096 samples, in powers of 2, and a slope window [23]. WMA uses between 256 and 4096 samples, in powers of 2, and an unknown window function.

All the formats use half-overlapping windows. Except for MP3 which uses a hybrid filter bank, all the formats use an MDCT as the time-frequency transform. For more details, the reader is referred to the provided references.

III. System

A. Algorithm

We propose a system which can identify from an audio recording that has undergone lossy compression the parameters used at the time-frequency analysis stage of the encoding, by measuring traces of compression, namely time-frequency coefficients quantized to zero, for different sets of parameters and at different framing positions in the audio signal. The idea is that the zeroed time-frequency coefficients will become visible only when the parameters and framing match those used for the encoding. Because different coding formats use specific compression parameters, we can consequently infer the correct lossy audio coding format.

Fig. 1 shows the overview of our audio compression identification system. Given a set of parameters associated with a known coding format, i.e., time-frequency transform t, window function w, window length N, hop size s, etc., the algorithm first derives the spectrograms in dB from segments $S_{i:L+i-1}$ of L samples, starting at successive samples i in an audio signal. It then computes the average energy over all the time-frequency coefficients for every spectrogram and takes the difference between successive overall energies. Assuming that the audio content stays the same between two adjacent spectrograms, the differences of energy will then be close to zero, except when the parameters and framing match those used for the encoding, in which case, zeroed coefficients will appear and a large difference in energy will be observed.

After keeping only the positive differences, the estimated maximum will then correspond to a matching score regarding the given set of parameters, while the corresponding index will indicate the position in the audio signal where the framing took place. In practice, peaks will be observed, typically every s samples, matching the hop size used at the time-frequency analysis stage, typically $N/2$. Unlike estimating the number of zeroed coefficients or computing the amplitude fluctuation in individual spectrograms, as presented in some previous works [3]–[6], computing the differences of energies between adjacent spectrograms was shown to be a measure more robust to changes in the audio content, leading to more accurate results, even with high bit rates and distorted analog audio.

B. Post-processing

We also propose a post-processing stage to refine the results obtained using the algorithm described above. Since we only need a short audio block (typically, around 1 second) to return meaningful estimates, we can combine the estimates from multiple blocks in a longer audio recording to improve the identification. The idea is to make use of the indices when combining the scores, as a correct match will show peaks happening at periodic locations in the signal (every s samples).

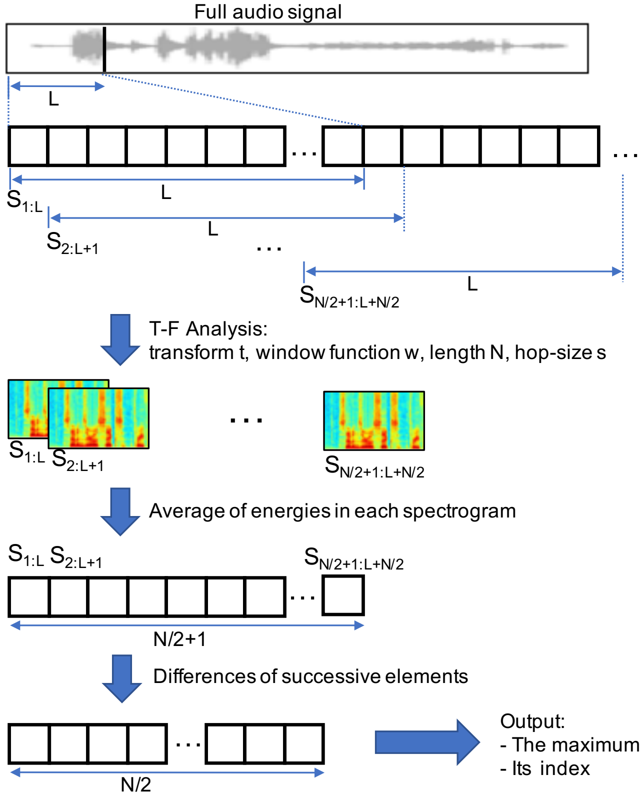

Fig. 1. Audio compression identification algorithm overview.

We first translate every pair of scores and indices obtained for an audio recording, and a given set of parameters, into polar coordinates; every score thus becomes a radius in dB, while every index is mapped into an angle in radians (modulo its periodicity). We then take the circular mean of those points, by converting them into Cartesian coordinates and computing their arithmetic mean. We finally obtain a single point whose radius will correspond to a final score, for a given set of parameters. The estimates can therefore be seen as points located on a circle; their centroid will have a large radius only if most of the points have large radii and similar angles. This simple method was shown to help deal with sample desynchronization, for example, introduced through analog transfer, improving the final identification.

Fig. 2 shows the results of our system applied to an audio example compressed using AC-3, and with analog transfer. The system analyzed 10 successive blocks from the audio example, using 5 different sets of parameters (t, w, N, and s) associated with the 5 popular lossy coding formats, i.e., MP3, AAC, AC-3, Vorbis, and WMA. As we can see (left plots), the set of parameters associated with AC-3 returned differences of energies (10 superimposed arrays) with high peaks located around the same position (modulo 256 samples, the hop size for AC-3). Peaks are not perfectly aligned because of the sample desynchronization introduced by the analog transfer. When mapped into polar coordinates (right plot), those peaks translate into points with large radii and similar angles; their centroid will therefore have a large radius, leading to a high final score. For the other coding formats, even if spurious peaks are observed, because of the irregular indices, they will translate into points scattered at different angles, therefore leading to centroids near zero and low final scores.

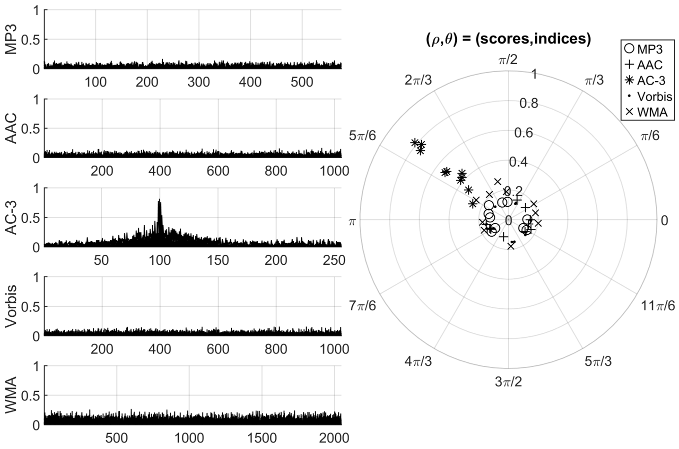

Fig. 2. Results for an audio example encoded with AC-3.

IV. Evaluation

A. Dataset

We propose to evaluate our system in identifying the correct coding format from audio recordings encoded using one of the five popular lossy audio coding formats, i.e., MP3, AAC, AC-3, Vorbis, or WMA, at different bit rates, including high bit rates, and with both digital and analog transfer, the latter scenario which will introduce sample desynchronization and electronic noise in the audio signal. We could not directly compare our system to previous works, since, as far as we know, this particular task has never been done before, previous works were not fully developed or tested with multiple coding formats, high bit rates, and distorted audio, and no relevant algorithm or dataset has been made available for comparison. Instead, we propose an evaluation of our system on a comprehensive dataset that we built as follows.

We first extracted one-minute audio excerpts from 10 various songs (from CDs) and 10 various movies (from Blu-rays). We obtained 20 different uncompressed stereo WAV files, sampled at 44.1 and 48 kHz for the song and movie excerpts, respectively. We then compressed these files using FFmpeg[1] with each one of the five popular lossy audio coding formats, again MP3, AAC, AC-3, Vorbis, and WMA, and at different bit rates, namely 96, 128, 192, 256, and 320 kbit/s. These compressed audio files were finally converted back to WAV files. The final audio recordings therefore do not have any metadata associated with their original coding format. In a real-world scenario, the audio recordings could be audio streams, for example, captured on a set-top box.

In addition to this set of digital audio examples, we also created a set of analog audio examples, by playing the digital audio through a phone connector and capturing the analog counterpart with a USB audio interface. This introduced distortions in the audio in the form of sample desynchronization and electronic noise which are meant to make the identification more challenging. In a real-world scenario, this can correspond to an audio stream captured through an analog output, for example, on a set-top box. We finally obtained a total of 1,000 one-minute audio examples, for a total of 16.7 hours of data.

B. Settings

We set our system to identify the five popular lossy audio coding formats described in Section II, by searching for the sets of parameters associated with them. For practical reasons, we fixed the time-frequency transform to an MDCT in all cases, even though MP3 actually uses a hybrid transform. We also used one fixed window length and one fixed window function for each coding format, ignoring the less common short and hybrid cases. We finally used a hop size equal to half the window length in every case. In summary, we tested MP3 with 1152 samples and a sine window, AAC with 2048 samples and a KBD window with α = 4, AC-3 with 512 samples and a KBD window with α = 5, Vorbis with 2048 samples and a slope window, and WMA with 4096 samples and a chosen sine window. Our initial tests showed that these settings were sufficient to return correct identification.

We ran our system on every audio example of the dataset described earlier, testing every set of parameters described above. After taking the average over the channels, we first segmented every one-minute audio example into blocks of one second, leading to 59 audio blocks per example (the last block being ignored). We then applied the algorithm described in Section III-A to every audio block. To minimize the impact of potential spurious peaks, we normalized the differences of energies using the standard score before searching for a maximum. We also discarded the audio blocks which returned very low scores (below a chosen threshold).

We finally applied the post-processing described in Section III-B to the remaining estimates and obtained a final score for every audio example, and for every coding format. To further improve the results, we multiplied the final score by the number of kept blocks, in practice leading to a circular sum rather than a circular mean. We considered a match if the correct coding format returned the highest final score, and a non-match if an audio example got all of its blocks discarded.

C. Results

Table I and II show the identification accuracy for the set of digital and analog audio examples, respectively. For each combination of coding format and bit rate, 20 audio examples (i.e., 10 song and 10 movie excerpts) were analyzed and 20 final scores were obtained. An accuracy of 0.9, for example, means that 18 out of 20 audio examples have been correctly identified, while 2 were non-matches. As we can see, the system was able to identify the correct lossy audio coding

[1] https://www.ffmpeg.org/

format in most of the cases, with an overall accuracy of 0.996 for the 500 audio examples of the digital set and 0.924 for the 500 audio examples of the analog set. With the settings of our system, we did not observe any misclassification.

TABLE I
IDENTIFICATION ACCURACIES FOR THE DIGITAL SET.

Digital	96k	128k	192k	256k	320k	all
MP3	1.0	1.0	1.0	1.0	0.9	0.98
AAC	1.0	1.0	1.0	1.0	1.0	1.0
AC-3	1.0	1.0	1.0	1.0	1.0	1.0
Vorbis	1.0	1.0	1.0	1.0	1.0	1.0
WMA	1.0	1.0	1.0	1.0	1.0	1.0
All	1.0	1.0	1.0	1.0	0.98	0.996

TABLE II
IDENTIFICATION ACCURACIES FOR THE ANALOG SET.

Analog	96k	128k	192k	256k	320k	all
MP3	1.0	1.0	0.95	0.75	0.7	0.88
AAC	1.0	1.0	1.0	1.0	0.8	0.96
AC-3	0.95	1.0	1.0	0.95	0.75	0.93
Vorbis	1.0	1.0	1.0	1.0	0.25	0.85
WMA	1.0	1.0	1.0	1.0	1.0	1.0
All	0.99	1.0	0.99	0.94	0.7	0.924

Note that the identification typically becomes harder when traces of compression are scarce in the audio signal, which is the case for high bit rates (e.g., 320 kbp/s) as less data is getting removed and for soft audio content (e.g., quiet scenes in movies) as there is less data to be removed. The identification is also more challenging for the analog set, mostly because of the sample desynchronization introduced by the analog transfer, which makes the framing detection harder, especially at high bit rates and with soft audio content. Despite of those limitations, our system was still able to identify the correct parameters in most of those hard cases.

We finally noted an overall negative correlation between the bit rates and the matching scores. Again, as the bit rate goes higher, traces of compression become scarcer, so our system returns a lower score. Future work will make use of this correlation to help estimate the bit rates as well.

V. CONCLUSION

We have proposed a system which can estimate from an audio recording that has undergone lossy compression the parameters used for the encoding and consequently identify the corresponding lossy coding format. We evaluated this system with audio excerpts from songs and movies, compressed into popular lossy coding formats, using different bit rates, and captured digitally and with analog transfer. Results showed that our system can identify the correct coding format in almost all cases, even at high bit rates and with distorted audio, with an overall accuracy of 0.96. Future work includes the development of a more efficient search algorithm, the identification of additional compression cues, and the application of the system to real-world data.

REFERENCES

[1] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards*. Springer US, 2003.

[2] W. R. T. ten Kate, "Maintaining audio quality in cascaded psychoacoustic coding," in 101th *Audio Engineering Society Convention*, 1996.

[3] J. Herre and M. Schug, "Analysis of decompressed audio - the "inverse decoder"," in 109th *Audio Engineering Society Convention*, 2000.

[4] S. Moehrs, J. Herre, and R. Geiger, "Analysing decompressed audio with the "inverse decoder" - towards an operative algorithm," in 112th *Audio Engineering Society Convention*, 2002.

[5] R. Yang, Z. Qu, and J. Huang, "Detecting digital audio forgeries by checking frame offsets," in 10th *ACM Workshop on Multimedia and Security*, 2008.

[6] D. Gärtner, C. Dittmar, P. Aichroth, L. Cuccovillo, S. Mann, and G. Schuller, "Efficient cross-codec framing grid analysis for audio tampering detection," in 136th *Audio Engineering Society Convention*, 2014.

[7] R. Yang, Y.-Q. Shi, and J. Huang, "Defeating fake-quality MP3," in 11th *ACM Workshop on Multimedia and Security*, 2009.

[8] B. D'Alessandro and Y. Q. Shi, "MP3 bit rate quality detection through frequency spectrum analysis," in 11th *ACM Multimedia and Security Workshop*, 2009.

[9] Q. Liu, A. H. Sung, and M. Qiao, "Detection of double MP3 compression," *Cognitive Computation*, vol. 2, 2010.

[10] T. Bianchi, A. D. Rosa, M. Fontani, G. Rocciolo, and A. Piva, "Detection and classification of double compressed MP3 audio tracks," in 1st *ACM Workshop on Information Hiding and Multimedia Security*, 1999.

[11] M. Qiao, A. H. Sung, and Q. Liu, "Improved detection of MP3 double compression using content-independent features," in *2013 IEEE International Conference on Signal Processing, Communications and Computing*, 2013.

[12] S. Hiçsönmez, H. T. Sencar, and I. Avcibas, "Audio codec identification through payload sampling," in *IEEE International Workshop on Information Forensics and Security*, 2011.

[13] S. Hiçsönmez, E. Uzun, and H. T. Sencar, "Methods for identifying traces of compression in audio," in 1st *International Conference on Communications, Signal Processing, and their Applications*, 2013.

[14] R. Korycki, "Authenticity examination of compressed audio recordings using detection of multiple compression and encoders' identification," *Forensic Science International*, vol. 238, 2014.

[15] D. Luo, W. Luo, R. Yang, and J. Huang, "Identifying compression history of wave audio and its applications," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 10, 2014.

[16] D. Seichter, L. Cuccovillo, and P. Aichroth, "AAC encoding detection and bitrate estimation using a convolutional neural network," in 41st *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[17] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Codec independent lossy audio compression detection," in 42nd *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[18] K. Brandenburg and G. Stoll, "ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, vol. 42, 1994.

[19] K. Brandenburg, "MP3 and AAC explained," in *AES* 17th *Conference High Quality Audio Coding*, 1999.

[20] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "ISO/IEC MPEG-2 advanced audio coding," *Journal of the Audio Engineering Society*, vol. 45, 1997.

[21] C. C. Todd, G. A. Davidson, M. F. Davis, L. D. Fielder, B. D. Link, and S. Vernon, "AC-3: Flexible perceptual coding for audio transmission and storage," in 96th *Audio Engineering Society Convention*, 1994.

[22] *ATSC Standard: Digital Audio Compression (AC-3, E-AC-3)*, Advanced Television Systems Committee, 2012, http://www.atsc.org/wp-content/uploads/2015/03/A52-201212-17.pdf.

[23] *Vorbis I specification*, Xiph.Org Foundation, 2015.