Audio Fingerprint Parametrization for Multimedia Advertising Identification

José Medina*, Francisco Vega†, Daniel Mendoza†, Víctor Saquicela† and Mauricio Espinoza†

*Department of Electrical, Electronic Engineering and Telecommunications, University of Cuenca, Ecuador
jose.medina@ucuenca.edu.ec

†Computer Science Department, University of Cuenca, Ecuador
{francisco.vegaz, daniel.mendoza, victor.saquicela, mauricio.espinoza}@ucuenca.edu.ec

***Abstract*—This article follows step by step a general framework for fingerprint extracting in order to develop a system for advertisements monitoring. The parameterization process uses some spatial and spectral characteristics measured over 600 advertisements that contain various types of sounds. Key factors such as accuracy, process time and granularity are analyzed together in order to enhance the system performance. At the end, the algorithm shows an accuracy of 99% using three seconds of granularity samples, and also the best compromise between time process and performance is achieved. This study suggests a set of parameterization steps which could be successfully implemented in other related audio applications.**

***Index Terms*—Automatic Content Recognition, audio fingerprint, advertising monitoring, signal processing.**

I. Introduction

Automatic Content Recognition (ACR) techniques have been developed with great interest due to its applicability in some areas [1], [2], such as broadcast monitoring, metadata collection and event flow synchronization. [3], [4] and [5] describe two important ACR technologies, *watermarking* and *fingerprint*. Watermarking is based on inserting an identifier embedded into the signal content in order to track individual signals characteristic. Even though, watermarking has high performance for signal post-processing and noise addition robustness, it is less used because it needs the manipulation of multimedia content. On the other hand, fingerprinting is a signal analysis technique that does not alter the original content. This technique is based on taking certain unique audio features, storing in a database and compare with a sample; a technique called Query by Example (QbE) [6].

Many algorithms are presented in the state of art for supporting fingerprint, but in a way a lot of them are variations of two father algorithms Shazam [7] and Phillips [8]. These two algorithms are focused on improving music information retrieval. However, music is just one important field in content analysis and fingerprint concept can easily apply to other contexts as advertisement monitoring. Nowadays, there is a great variety of multimedia advertising contents that are scattered on various media, such as Radio, Television and Internet. The information generated by multimedia content is closely linked with media marketing which uses that in order to obtain a general idea of how products and services are positioned to market. Furthermore, a few years ago, countries like Ecuador have adopted laws for monitoring contents [9], which allows the government adequately manage the radio and television concessions. In many cases the processes of monitoring are manual and consumes an important amount of human resources. For this reason, it is important to concern about ACR of advertisements.

Advertisements has different features than music because it also contains speeches and multiples kind of noise. So, those methods that have good results in music may fail searching advertisements. There are important features that difference speech than music, for that, an advertisement fingerprint should consider another way of parameterization, and keeps the same results for audio degradation. Thus, advertisement fingerprint could use not only for monitoring systems, but also for event flow synchronization in order to recognize adds and displays information with second screen devices [3] [4].

The rest of this paper is structured as follows: Section II shows and discuses related works about audio fingerprint. The general framework for fingerprint extraction is presented in Section III. Parameterization methodology is described in Section IV and Section V, focus on the evaluation and results on the implementation of advertising audio fingerprint. Finally, Section VI is reserved for conclusions and future works.

II. Related Works

This section summarizes the more relevant approaches related to fingerprint modeling techniques. In most cases fingerprinting follows a general framework as proposed in [1] and [2]. This framework is composed by two main components ”Fingerprint Extraction” and ”Search and Match” as shown [7] and [8], which are pioneer researches in the field. Also, different authors [1], [2], [7] and [8] identify some characteristics that should be embedded in the framework, such as:

- Robustness: A fingerprint system should allow identifying degraded audio samples.
- Accuracy: It is the system's capacity for retrieving a correct result.
- Reliability: This measures false positive and negative rates and how often the result is the correct.

978-1-5386-3894-1/17/$31.00 © 2017 IEEE

- Granularity: The audio sample used for recognition should be very small around few seconds (3-10 seconds).
- Storage: Only essential audio features must be stored.
- Fastness: Search speed should be in the order of milliseconds for largest content database, and it should be implemented with limited computational resources.
- Scalability: The audio fingerprint system must allow to increase the database considering millions of audios.

Some optimizations in one of these characteristics are presented in many related works. For instance,[8] describes a parameterization method that focus on small granularity; algorithm's reliability also is improved developing a new method for rejecting true positives calculating the Bit Error Rate (BER) threshold. In [10] fingerprint modeling is related to robustness avoiding audio degradation causes for time-stretching. Another technique proposed in [7] focus on robustness, the recognition method allows to work with audio degradation. Also, storage is reduced due to fingerprints are composed only by two points. Acoustic feature extraction is based on frequency salient points that keeps invariant for postprocessing. In [7] and [11] the authors consider peak points with high energy because these keep invariant to degradation. So, some audio peaks also appear in a degraded audio sample allowing a successful comparison. Works as [8], [12], [13] and [14] use the same technique, but they also consider the variation of energy in spectrogram zones. Energy peaks and zones of energy variation can reject the harmonics generated by noise due to these harmonics only appear in some points and do not affect the total energy zone. Fingerprint modeling and search process are based on two models proposed by [7] and [8]. The model of [7] uses a hash table which is indexed taking two peaks of energy in the spectrogram (fixed point and a target zone point). In the search method a function looks for audio candidates that have the same peaks components. [13] adds the distance between peaks and energy variation zones as a variation of this method. The new index is used for reducing collisions in the hash table reducing the false positive rate. The second model is explained in [8], he proposes to generate a sub-fingerprint with a binary code of 32 bits per time frame. The sub-fingerprint is build taking a threshold of power in a frequency component. The search process for this model takes a block of sub-fingerprints and looks for those candidates in the database whose bit error rate (BER) is lower than a threshold. A variation of this method is presented in [15], where the main change consists in a sort the sub-fingerprint sequences in order to accelerate the process of candidate selection.

III. General Framework for Fingerprint System

As was introduced in the previous section, a general framework to support audio fingerprint is composed of two main components: "Fingerprint Extraction" and "Search and Match". This section describes these components in detail.

A. *Fingerprint Extraction Component.*

This component consists of a Front-End and a Fingerprint Modeling module that transform audio signals to a set of acoustic features and combining these in order to build a fingerprint. Front-End must cover various characteristics, such as robustness, optimal storage, and granularity. On the other hand, the Fingerprint Modeling combined to Search and Match processes should looks for an optimization in accuracy, reliability and search time.

1) Front-End: The fingerprint extraction process should follow an ordered set of steps in order to model a fingerprint.

- **Preprocessing:** This step is focused on normalizing audio files because there are a lot of audio sources with different sampling rates, compression codecs and formats.
- **Framing and Overlap:** In order to extract temporal features the audio should be divided into a number of frames. Each frame is separated and analyzed independently, but overlapping is necessary because two sequential frames must have a high correlation.
- **Spectral Transform:** Audios are transformed to spectral signals, increasing the number of acoustic features. Transformation is related to the use of computational resources, so generally Fast Fourier Transform (FFT) is used.
- **Feature Extraction:** Time and spectral characteristics are taken for generating a vector of acoustic features. There are a variety of techniques [7] takes out highest energy points, [8] and [13] takes energy variations or the features are extracted from a spectral band analysis [11].
- **Postprocessing:** This step reduces computational resources or increment robustness in the fingerprint. For example, acoustic vector can add a new feature in order to improve collisions in search.

2) Fingerprint Modeling: This process receives all acoustic features extracted in the previous step and builds a fingerprint representation. There are multiple ways for fingerprint modeling, some of them take distance metrics and others are based on indexing the acoustic features in a binary representation. Two widely used techniques are explained below.

- **Technique based on distance metrics:** The work [7] proposes to generate metrics based on two fixed points in the spectrum, these two points are an anchor point and a point in a target zone. The final result is a set of index values that contains three variables: frequency of the first point, frequency of the second point, and time distance between both points. There is another inherent parameter which should be configured, it is the distance between a point and the beginning of the target zone. For instance, a global audio fingerprint can take points more separated, on the other hand, a very granular fingerprint should take nearby points. A variation of this technique is explained in [16], in this case the author builds two quadratic zones in order to generate a more robust fingerprint, another author [13] proposes to add a third fixed point for reducing false positive rate.
- **Technique based on sequences:** The first version of this technique is explained in [8]. It consists in generating a sequence of binary sub-fingerprints. The audio spectrogram is divided into 32 frequency bins in logarithmic

scale. Each bin is evaluated as one or zero comparing its energy with its neighbor bin energy. Finally, a database is created taking each binary sub-fingerprint as the key value of a table and the values stored in the table are the sub-fingerprint position in the audio. The position will help later to rebuild sub-fingerprint block and compare a original audio with a audio sample.

In the following the second component of the general framework to support audio fingerprint is explained.

B. Search And Match Component

The Search and Match block is focused on searching candidates in the database, discriminating and retrieving a result as fast as possible. Based on the way that fingerprint was modeled, two techniques can perform this task.

- **Search based on hash tables:** The model proposed in [7] uses this technique for building a hash table whose index is formed by three distance metrics. The key of search is that the hash table has many collisions and various audios can have the same index, for that reason the hash table to retrieve a set of candidates for each index. Different index values did not retrieve the same candidates, then a ponderation can be easily made looking for that audio which appears more frequently. Even though the search method can retrieve a correct audio, the process is not enough for degraded audios where the noise can generate many candidates for each index. For that, a matching method is used for rejecting true positives and finds the correct one. This technique estimates the time offset of the audio sample in the original audio. An index value can retrieve different offset for each candidate, but one candidate keeps the same offset for all index values. Using a histogram the match technique can find the correct candidate and discriminate others.
- **Search similarity between sequences:** This technique allows to search over fingerprint made by binary sequences. It focuses on calculating similarity between two blocks of fingerprints through indicators as BER or Longest Common Subsequence (LCS) that are compared to a threshold. The compute of the threshold is very important because it has a direct impact on accuracy. For example, a small threshold can increase the false negative rate and a high threshold can produce a high false positive rate. The number of bits in a block has a direct impact in the search time, for that, the search and match method should work considering blocks of small granularity.

These two techniques can be combined in order to improve the search time. For example, in [8], [10], [15] and [12] a hash table is used for looking for candidates, then a method of similarity is applied for reject candidates.

IV. Parametrization Methodology

The parameterization criteria for the general framework are described in this section; the configuration is made for advertisements recognition. So, the parametrization is based on statistical metrics measured over 600 advertisements.

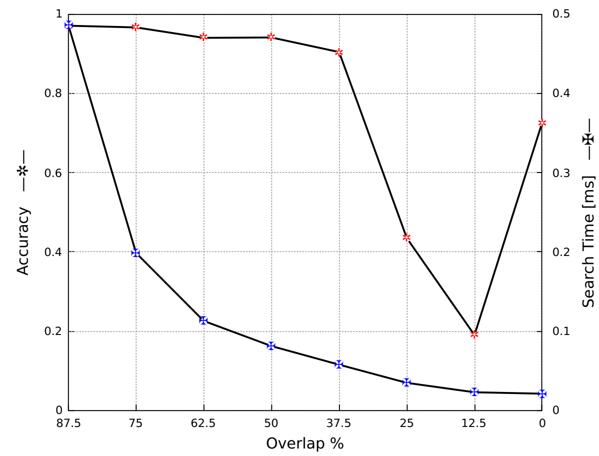

Figure 1. Overlap effects over search time and accuracy

A. Front-End

The metrics in the Front-End are described below.

- **Preprocessing:** In this step, audio files are digitalized with a general format that consist in 8Khz for the sampling rate, raw format audio, mono channel and uncompressed codec. The low sampled rate was chosen because the system can be used for QbE [7], where a poor quality audio is recording and searching. A 8Khz sampling rate implies that frequency components are lower than 4Khz, these are enough for extract voice and some musical features.
- **Framing and Overlap:** The audio data is divided into frames of 8 milliseconds. Each frame is windowing with a hamming window in order to remove harmonics generated by discontinuities. The accuracy is affected by the correlation between two contiguous frames, for that, overlap is the main parameter in this step. Figure 1 shows a comparison of accuracy and search time that helps to choose an adequate overlap. Overlap affects the total number of frames, then the algorithm can spend more time in searching. Based on the figure, if an overlap greater than 50% was chosen it would increase six times the search time. Thus, an overlap = 50% where the accuracy is greater than 95% is enough and it can increase later with other operations.
- **Transform:** All audio files are transformed using FFT. 512 logarithmic frequency bins are computed for each temporal frame. Hence, 4khz bandwidth generates bins of 8hz in amplitude. As a result, transformation have a spectrogram with 3 components: time frame, frequency bin and spectral energy.
- **Feature extraction:** This process consists in extracting peak points (coordinates in time and frequency). A first consideration in this phase is remove low frequency components around 800Hz, Figure 2 shows this band that has a concentration of energy with many peak points, it can increase the false positive rate when fingerprints are compared. Figure 2 also shows that overlapping and FFT

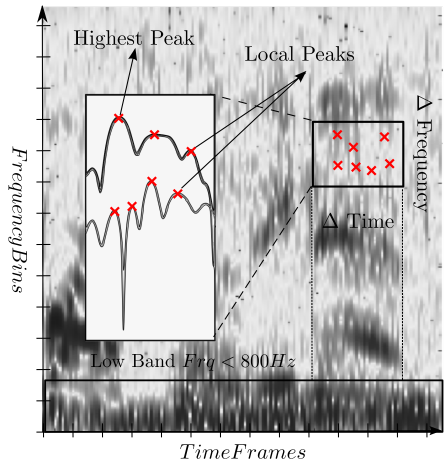

Figure 2. Local peaks and spectral energy concentration

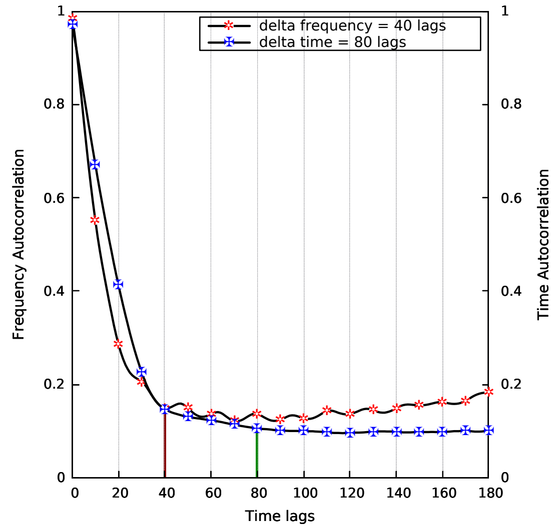

Figure 3. Time and Frequency Correlogram for Ads Data Set

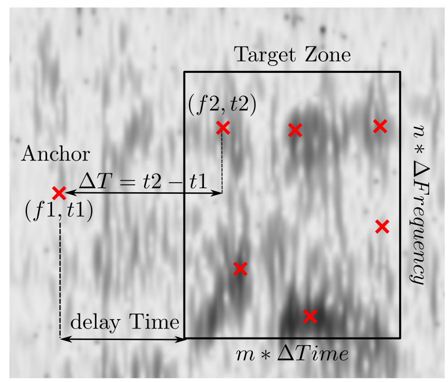

Figure 4. Anchor point and target zone parameters

generate contiguous peaks in spectrum (local peaks). In order to avoid local peak extraction a window zone is chosen based on the analysis of autocorrelation in time and frequency as shown in Figure 3. This correlogram establish the minimal number of frames and bins where the correlation is lower for considering independence in time and frequency. So, 40 frequency bins are chosen because in this point the plot decrease until 0.15, then it keeps an oscillation. Meanwhile, 80 frames is the independent time point because the correlation decrease and then keeps invariant.

- **Postprocessing:** Considering all peaks for modeling can increase the total number of fingerprints. Postprocessing reduces a percentage of peaks, it affects the probability of a peak in the spectrogram and is analyzed taking into account the fingerprint modeling in the next section.

B. Fingerprint Modeling

The fingerprint modeling is based on two points of reference as shown in [7]. An anchor peak and a target zone peak are combinatorial associated (see Figure 4). The metrics of time and frequency in each two points generates an index that contains three values: anchor frequency, second point frequency and time distance (f1,f2,ΔT). The target zone is parameterized considering a delay time and its size.

- **Delay Time:** It is the minimum time distance between an anchor point and the target zone in order to consider these as independent events. Delay time is established using the correlation between a frame and the n contiguous frames. For this case the frame 200 (0.8 seconds) is the point where the correlation is zero, from there an anchor point has no incidence on target zone.
- **Target zone size:** As shown in Figure 4 the target zone has a size of $m * \Delta Time \times n * \Delta Frequency$, but in this case a quadratic target zone is considered that means $m = n$, then the equation 1 is analyzed.

$$P_F = (1 - (1-p)^F) \tag{1}$$

The equation expresses that with F as number of peaks in the target zone and p as the probability of a peak in the spectrogram, then the probability of at least one peak surviving in a target zone is $P_F = (1 - (1-p)^F)$.
In order to find an optimal F and p the probability of at least one point surviving is established with a high value $P_F = 0.98$. In postprocessing a reduction of peaks was considered affecting p. Figure 5 shows the balance in probability, search time and granularity. The figure exhibits an inverse effect comparing p and search time because small number of peaks implies an increment in F (wide target zone) in order to maintain $P_F = 0.98$. Also, a greater target zone implies increasing in granularity because it is necessary at least an anchor point and one target zone for building a fingerprint.
In the Figure 5, the optimal peak reduction is established in $p = 0.7$ because it keeps a low granularity and search time is very low (0.6 ms) with an accuracy greater than 98%. The value of p allows to estimate an optimal $F = ln(1-0.98)/ln(1-0.7) \approx 4$. Also, the probability $p = 0.7$ establish that the system needs 6 quadrants zones

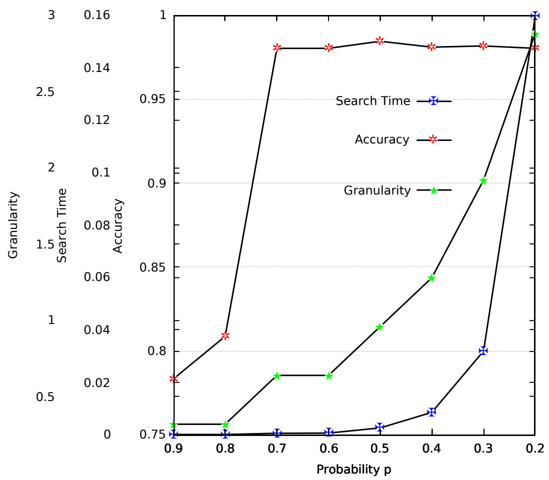

Figure 5. Effects of peaks reduction

of size $\Delta T \times \Delta F$ in order to generate 4 peak points in the target zone ($4/6 \approx 0.7$). Hence, the target zone size is calculated by $n = \sqrt{6} \approx 3$.

C. Search and Match

Search consist in a process for building tables and improve the speed for retrieving candidates. Match focus on ranked candidates in order to retrieving an optimal result. Section IV-B explained that each fingerprint generates an index of $[f1, f2, \Delta T]$, this index was used to build a hash table easily implemented as Look Up Table (LUT) described in [8]. The values stored in the table for each index are the audio identificator and the anchor time value called offset as described in [7]. This table is used for explaining the search and match process below.

Search: This method consists in load the LUT in RAM memory then search a set of audio sample fingerprints. A search can retrieve various candidates in a vector. The data in the vector are composed of two values (audio id , offsets) as shown in Figure 6. Candidates should be processed by a match method which determines the correct audio and reject audios whose points were found only by audio degradation.

Match: Match process uses two ponderation techniques as shown in figure 6. The first one consists in to separate each value of the vector, and then sort these by audio ids. The candidates are ranked by the number of found points. In this step some of them can be rejected because their numbers of points are not enough for recognition. This ponderation keeps a high recall because the correct candidate always appear ranked in the top positions and according to the application of the system this method could be sufficient for retrieving a result. The second ponderation takes into account the offset value as shown [7]. The original and sample audio offsets are subtracted in order to find the displacement of the sample into the original audio. As seen in the Figure 6, if the points were original peaks, their offsets subtraction would retrieve more points with the same displacement value. All candidates can have several separation distances, so the points which generate dispersion in distance are eliminated and the equal separation points are chosen as true points. The candidates are ranked again and now the system can retrieve a result more accurately.

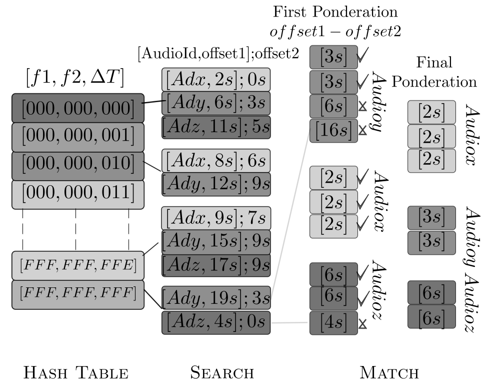

Figure 6. Fingerprint Search and Match.

V. Evaluation and Results

The evaluation process consists in comparing an algorithm parameterized according this approach with two fingerprint systems. The first system is a toolkit called Dejavu available in [17]. The second one is an industrial service provided by ACRCloud [18]. The proposed algorithm and Dejavu are implemented using Python in a computer with basic resources (Ubuntu 16, Intel core i7 3.4Ghz and 8Gb RAM) and ACRCloud uses the could services provided by [18].

Table I shows the results for a monitoring scenario because it evaluates these three systems over random noiseless audio samples of 1, 2, 3 and 5 seconds. Underprivileged ACRCloud account has some restrictions in minutes of content, for that reason, the evaluation was made over 300 minutes of content over 363 advertisements data set. All systems consider a successful result when the original audio is found, a false positive is when the system retrieves various results or a different audio and a false negative is when the system does not retrieve any result. Results establish that:

- The proposed algorithm is not suitable for granularity=$1s$ because the system was parameterized considering a delay time $0.8s$ and a size of target zone $0.96s$. Thus, the system is effective for a minimum granularity $1.76s$. So, when the sample granularity increases to two seconds the algorithm improves significantly to 98.9%.
- Although, Dejavu and ACRCloud returns a good accuracy greater that 84% using one second sample, these can not produce a notably increment for samples of two, three and five seconds when the maximum accuracy is 99%. Meanwhile, the proposed algorithm can be achieved a 100% success rate in just five seconds.
- Dejavu and ACRCloud have a higher false positive rate compared to the proposed algorithm for granularity $> 2s$ because two different advertisements can share the musical background and it produces ambiguity in the response. The proposed algorithm can discriminate false positives

Table I
RESULTS COMPARING THE PROPOSED ALGORITHM (PA), DEJAVU ALGORITHM (DJU) AND ACRCOULD ALGORITHM (AC).

	PA	Dju	AC	PA	Dju	AC
Granularity	1 second			2 seconds		
Recognized	12	313	307	359	347	359
False Positive	351	0	54	4	2	0
False Negative	0	50	2	0	14	4
Accuracy	0.033	0.862	0.846	0.989	0.956	0.989
Process Time ms	1.50	120	—	3.01	247.8	—
Granularity	3 seconds			5 seconds		
Recognized	362	359	360	363	361	362
False Positive	0	0	0	0	0	0
False Negative	1	4	3	0	2	1
Accuracy	0.997	0.989	0.0.991	1.00	0.994	0.997
Process Time ms	5.19	339	—	9.2	543.5	—

due to it takes out not only features related to music, but also features related to voice.

- The storage is shown as another indicator for robustness, it can be only compared between the proposed algorithm and Dejavu. The proposed algorithm was parametrized in order to maintain a good compromise on the number of features extracted and the system performance. For that reason, the system only needs 16.1 MB for storage 300 minutes of content. On the other hand, Dejavu takes out a lot of features in order to maintain the accuracy. That is evident because Dejavu needs two channels of audio and the total database storage is around 470 MB. It has a direct impact in the search time that is shown below.
- Although the systems are not evaluated over a large-scale data set, search time is another important indicator. ACRCloud is implemented as a cloud service; so, its time is not comparable with a system evaluated in a computer with basic resources. However, the proposed algorithm and Dejavu use the same computational resources; Thus the proposed algorithm proves that is fast and very scalable because its total process time is very low with a maximum of 5.19 milliseconds compared to the Dejavu algorithm that retrieve a response in 543 milliseconds.

A demo is shown in http://190.15.132.90:9051/fingerprint.

VI. CONCLUSIONS AND FUTURE WORKS

This work has described a parameterization process for an audio fingerprint system. The parameterization focus on advertising and all features were extracted considering parameters inherent to advertisements like voice, noises and music mixes. The proposed parameterization was tested on a monitoring scenario with an accuracy higher than 99% for a granularity greater than 3 seconds. Furthermore, it proves to be just as efficient as an industrial application for audio recognition. The final algorithm shows to maintain the compromise between characteristics as:

- Search Time: Few acoustic points allows accelerating search time, these will increase for degraded audio applications because these need to consider more points.
- Granularity: The parameterization achieves a successful recognition with a small sample.
- Accuracy: The fingerprint shows a high accuracy for few points, but it could be improved considering new methods for ranking candidates.

Finally, this work proves that a general method of fingerprint extracting can be improved parameterizing features according to a specific scenario. As future work, the application of the system can also extended for QbE because some considerations implies to have degraded audio, but it could imply extend the granularity or develop a new method to match the results.

VII. ACKNOWLEDGMENTS

The work presented in this article is part of a research project called “Use of semantic technologies to analyze multimedia content broadcasted by digital television” supported by the Department of Research of the University of Cuenca (DIUC).

REFERENCES

[1] H. Kekre, N. Bhandari, N. Nair, P. Padmanabhan, and S. Bhandari, “A review of audio fingerprinting and comparison of algorithms,” *International Journal of Computer Applications*, vol. 70, no. 13, 2013.

[2] P. Cano, E. Batle, T. Kalker, and J. Haitsma, “A review of algorithms for audio fingerprinting,” in *Multimedia Signal Processing, 2002 IEEE Workshop on*. IEEE, 2002, pp. 169–173.

[3] “Automated content recognition creating content aware ecosystems,” CIVOLUTION, Tech. Rep., 2013.

[4] R. Mendes Costa Segundo and C. A. Saibel Santos, “Second screen event flow synchronization,” in *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1–7.

[5] D. Milano, “Content control: Digital watermarking and fingerprinting,” *White Paper, Rhozet, a business unit of Harmonic Inc*, vol. 30, 2012.

[6] A. Wang, “The shazam music recognition service,” *Communications of the ACM*, vol. 49, no. 8, pp. 44–48, 2006.

[7] A. Wang *et al.*, “An industrial strength audio search algorithm.” in *ISMIR*, 2003, pp. 7–13.

[8] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system.” in *ISMIR*, vol. 2002, 2002, pp. 107–115.

[9] A. N. del Ecuador, “Ley orgánica de comunicación,” 2013.

[10] J. George and A. Jhunjhunwala, “Scalable and robust audio fingerprinting method tolerable to time-stretching,” in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 436–440.

[11] H. Wang, X. Yu, W. Wan, and R. Swaminathan, “Robust audio fingerprint extraction algorithm based on 2-d chroma,” in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*. IEEE, 2012, pp. 763–767.

[12] D. Sui, L. Ruan, and L. Xiao, “A two-level audio fingerprint retrieval algorithm for advertisement audio,” in *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 2014, pp. 235–239.

[13] T. Jie, L. Gang, and G. Jun, “Improved algorithms of music information retrieval based on audio fingerprint,” in *Intelligent Information Technology Application Workshops, 2009. IITAW’09. Third International Symposium on*. IEEE, 2009, pp. 367–371.

[14] J. Shi, X. Yu, H. Liu, and W. Xiong, “Audio fingerprinting based on salient points for audio retrieval,” in *Smart and Sustainable City 2013 (ICSSC 2013), IET International Conference on*. IET, 2013, pp. 352–355.

[15] Q. Xiao, M. Suzuki, and K. Kita, “Fast hamming space search for audio fingerprinting systems.” in *ISMIR*, 2011, pp. 133–138.

[16] M.-L. Bourguet and J. Wang, “A robust audio feature extraction algorithm for music identification,” in *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

[17] WILLDREVO, “Audio fingerprinting with python and numpy,” http://willdrevo.com/fingerprinting-and-audio-recognition-with-python/.

[18] “Acrcloud,” https://www.acrcloud.com/, ACRCloud.