UC | Chile

International Computer Music
Conference 2021 Santiago, Chile

Proceedings of the International Computer Music Conference 2021 (Draft version)

The virtuoso computer: redefining limits

Hosted by:

Pontificia Universidad Católica de Chile
Santiago, Chile, July 25th−31st

Proceedings of the International Computer Music Conference 2021

Pontificia Universidad Católica de Chile | July 25th-31st, 2021, Santiago, Chile

The virtuoso computer: redefining limits

Rodrigo F. Cádiz, editor

ISBN: 978-0-9845274-9-6

Published by

International Computer Music Association, Inc.
1819 Polk St., Suite 330
San Francisco, CA 94109
USA

icma@umich.edu

http://www.computermusic.org

UC | Chile

International Computer Music
Conference 2021 Santiago, Chile

Using tf-idf and cosine similarity in Shazam-based fingerprinting algorithms

Arthur P. M. Tofani
Institute of Mathematics and Statistics,
University of São Paulo (IME-USP)
tofani@ime.usp.br

Marcelo Queiroz
Institute of Mathematics and Statistics,
University of São Paulo (IME-USP)
mqz@ime.usp.br

ABSTRACT

This study discusses the application of text retrieval techniques in Shazam-like fingerprinting algorithms. The goal is to filter query input fingerprints using tf-idf weights as a measure of relevance in order to reduce the number of records returned by the database. As accuracy could potentially be affected by this filtering approach, we investigate these requisites together, by looking for a filtering threshold τ that produces reduced database response payloads with minimal impact on accuracy. Furthermore, we also discuss the use of cosine similarity as an alternative to Shazam's original scoring method, in order to improve robustness against time distortions. We apply these techniques on three different datasets and discuss their benefits over the original method.

1. INTRODUCTION

Acoustical fingerprinting systems are often used in audio identification tasks. Even though many different systems have been proposed in the past decade, they commonly derive from one of two seminal works: Haitsma & Kalker (2002) [1], also known as the *Philips* fingerprinting algorithm and Avery Wang's original description of the *Shazam* algorithm (2003) [2]. These systems can be compared and assessed by various requirements such as accuracy, reliability, fingerprint size, granularity, speed, scalability and robustness.

Robustness is a particularly important requirement in the context of this work, and it is related to the ability of the algorithm to identify an audio clip after several signal degradations like compression artifacts, environment noise, tempo changes and others. These systems are known to provide satisfactory results on detecting identical copies of a same audio record. Yet, many works [3, 4] discuss the applicability of fingerprinting techniques in other less specific contexts such as version/cover songs identification.

The current work is part of a series of investigations towards the usage of such techniques for cover song identification (CSI) tasks, particularly focused in the Shazam-based family of algorithms, while handling two important drawbacks regarding the original algorithm's description that are responsible to make it unsuitable for CSI as well as many other lower-specificity tasks. These two drawbacks are detailed in the sequel.

1.1 Database response payload size

In the query phase, fingerprints are produced from the sample audio and then are matched against a database that usually contains a large amount of audio clips indexed by their own fingerprints. It means that a same indexed fingerprint in the database may relate to many different audio clips. Moreover, fingerprints are extracted by locating and combining local maxima events in the STFT magnitude spectrogram.

In the CSI context, it makes sense to use higher-level features (such as chroma features or PCP) instead of the spectrum directly, since this task regards to musical similarities rather than spectral ones, as seen in [3]. This approach may reduce the entropy of generated fingerprints, which may produce a large amount of candidate audio clips to be returned from the database. This drawback may not be readily observable when fingerprints are generated from the spectrum since the probability of collisions is lower. When using higher-level features, however, it becomes a critical drawback.

1.2 Low robustness against temporal distortions

Both *Philips* and *Shazam* algorithms offer very low robustness against temporal distortions such as time compression/stretching [5]. This is specially relevant in the CSI task context, since two different versions of a same song are not likely to share the same time patterns.

In the original description of Shazam algorithm, there are two implementation characteristics that decrease the robustness against temporal distortions. Firstly, the generated hashes are based in the time difference between two local maxima events. As a consequence, the same audio clip distorted in the time, even slightly, may produce completely different hashes since the time distance between the same events will produce a different result. Secondly, at the scoring phase, the algorithm evaluates the candidate audio records according to the frequency of time offsets between query and database samples. For an efficient scoring procedure, the author assumes that both query and the correct candidate will have the matching hashes in corresponding time-frequency relative positions.

This study investigates the application of tf-idf measure (described in the next section) as an approach to handle the previously mentioned disadvantages. Our main goal is to reduce the response payload of a given query by eliminating the non-relevant terms, or in other words, by removing the hashes that are present into many audio clips, whereas the hash key relates to many records into the inverted-index database table; all of these related records figure in the response payload (increasing its size) but few of these audio clips share a significant amount of matched hashes along with the query, and hence they're usually dropped off in the algorithm's scoring phase because of their low score values. Tf-idf measure has been already explored to handle content-based audio retrieval tasks by Riley et al. [4]. In their work, however, the authors used different approaches, with Locality Sensitive Hashing over full audio clip queries, with focus on evaluating robustness rather than the database response size.

2. METHODOLOGY

This study proposes to explore two different approaches supported by tf-idf calculations over the original Shazam's algorithm description. In the first approach, the goal is to eliminate non-relevant hashes from the database results and evaluate how it impacts in the response time and size, as well as the accuracy. In the second approach, the goal is to evaluate the applicability of the cosine similarity method to score the candidate results in place of the original method mentioned by Wang. Detailed procedures for both approaches are described below:

2.1 Applying tf-idf weights over fingerprints

Tf-idf (Term Frequency-Inverse Document Frequency) is a statistical measure that is intended to represent the relation between the relevance of a term in a given document in comparison to its relevance in a collection of documents or corpus [6]. This technique is widely used by search engines as they need to rank results by their relevance to a query phrase or expression; in this scenario we can expect that the search engine handles an enormous document collection and that some terms in the query might be very common among the whole collect (for example the article "the" or the pronoun "it"). *Term frequency* $\mathrm{tf}_{t,d}$ denotes the number of occurrences of a term t in a document d [7]. The document frequency df_t defined to be the number of documents in the collection that contain a term t. The *inverse document frequency*, idf_t, represents the frequency of term t in all documents in the collection. Also, this measure can be weighted by many approaches, and by the purposes of this study, the formula used is:

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t} \tag{1}$$

where N is the total number of documents in the corpus. We calculate tf-idf as the product of these two measures:

$$\mathrm{tfidf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t \tag{2}$$

Firstly, the database has to be adapted to compute $\mathrm{tf}_{h,d}$ and df_h for each incoming hash h (the fingerprint hashes will be the *terms*, and the audio clips will be interpreted as *documents*). Then, a threshold τ is added as a parameter, and its value is expected to be received along with the query; any occurrence of h with $\mathrm{tfidf}_{h,d} \leq \tau$ will be ignored by the database and hence these terms will not compose the query results.

2.2 Scoring query results with cosine similarity

Cosine similarity is widely used along with tf-idf to rank text results, and we will test its behavior in this particular scenario. It is worth mentioning that this ranking measure do not consider the time offsets; instead, this is based in the terms relevance, despite of their order of occurrence within an audio clip. Our expectation by changing the original algorithm's scoring method is to overcome one of the implementation characteristics that turns the original Shazam algorithm unfit for searching audio clips with time distortions. Given two documents d_1 and d_2 the cosine similarity of their vector representations $V(d_1)$ and $V(d_2)$ is stated by:

$$sim(d_1, d_2) = \frac{V(d_1).V(d_2)}{\|V(d_1)\|\|V(d_2)\|} \tag{3}$$

where the numerator represents the dot product of $V(d_1)$ and $V(d_2)$ [7], while the denominator is the product of their euclidean lengths.

Four different scoring strategies are applied to provide a better comparison: the original Wang's scoring strategy [2] (WNG), the cosine similarity (COS), the simple matching hashes count (CNT) and combination of the first two strategies (WNG+COS). The simple hash count is not a good strategy in practice since neither the hash positions nor relevance are considered, and hence CNT will be used for a lower-bound analysis of the scores. On the other hand, a combination of WNG and COS can produce better outcomes than each one separately, since these measures relate to different similarity aspects. This combination was done by re-scoring the outcomes from WNG and COS using *borda count* [8].

3. EXPERIMENT

The methods described in the previous section were applied in Audfprint fingerprinting algorithm [9], an implementation of the original Shazam's description. Three different datasets were used in this experiment, GTZAN [10] and FMA (Free Music Dataset) [11] and an extended version of Youtube Covers dataset [12] (henceforth YTC+). The first two datasets consist in 30 seconds music clips of various genres; The YTC+ is composed by full-length songs extracted from Youtube music videos. The dataset sizes are presented in Table 1. The experiment was performed using Python and the inverted index was implemented in a Redis database [1] .

The first step of this experiment consisted in adding new entries from the dataset to the database and calculating $\mathrm{tf}_{h,d}$, df_h and $\mathrm{tfidf}_{h,d}$ for every incoming hash h of an audio clip d. In the next step, we perform searches in the fingerprint system using query samples extracted from the same dataset. Each query is composed by the hashes extracted from the sample audio clip (H_q) and a threshold value τ between 0 and 30. When submitted to the system, we expect to receive the response R containing all candidates retrieved from the database, each candidate consisting of a tuple with an audio clip identification r and the set of timestamped hashes H_r that matched hashes in the

[1] The code is available in the URL https://github.com/arthurtofani/icmc2021-fingerprinting-evaluation

query. Then, by iterating over different values of τ, we perform searches passing $(H_{q,\tau})$ (i.e, all hashes in $H_{q,\tau}$ such that $\mathrm{tfidf}_{h,q}$ ¿ τ) and receive a set R of candidates (r, H_r). For each candidate, we calculate a score s_r through a given score function. The elements in R are ordered by this score value. The accuracy can be measured by picking the response in the first position, thus the record with greater score. For this experiment, we used queries of different sizes (3, 6, 9, 12 and 15 seconds), each group composed of 120 queries (40 of each size) per value of τ, all queries corresponding to one (and only one) audio clip in the database.

In the evaluation step, we addressed the two approaches described in the previous section separately. First, we evaluate the impact of different values of τ over the amount of results returned by the system $|R|$. It is expected that as τ is increased, the system returns less candidates by filtering those ones whose relevance of the matched terms are below the threshold value. Also, this procedure can impact in the system accuracy. To observe this impact, we also evaluate the average of top-1 scored audio clips that correctly match the query sample for each value of τ. Our goal is to find $\hat{\tau}$ such that we have the best average accuracy with the smaller $|R|$. It is worth mentioning that the comparison of the results with the original algorithms (without tfidf calculations) can be achieved by setting $\tau = 0$ (henceforth τ_0), since no hashes will be ignored in the database for this setting.

The second approach investigates the substitution of the scoring procedure originally presented [2] by the cosine similarity between H_q and H_r, since this method is based in the term's relevance instead of the term's time position in the audio clip. After querying, for each candidate in R, we compute s_r by using the cosine similarity method. In order to move towards a better understanding of the role of the scoring function, four methods to rank results in R will be evaluated: (a) the original histogram of time-offsets [2], (b) the cosine similarity, (c) a simple hash count method ($|H_r|$) and also the combination of a and b methods, considering that they classify the same input over different perspectives (temporal alignment and relevance), and that they could be combined to increase accuracy.

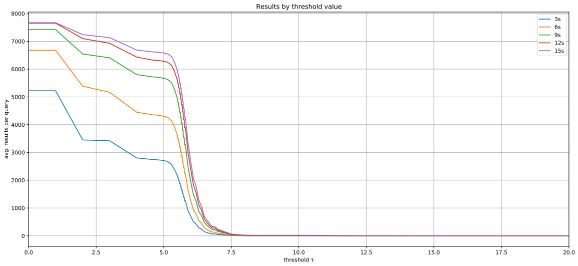

Figure 1. Average results returned by DB in terms of τ for different query sizes.

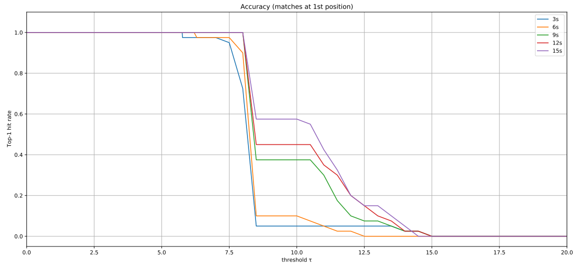

Figure 2. Accuracy in terms of τ (query sizes from 3 to 15 seconds)

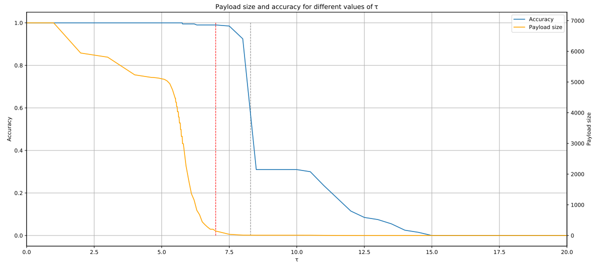

Figure 3. Payload size and accuracy values for different values of τ (for all queries up to 9 seconds in the FMA dataset. Red line indicates the selected $\tau = 7.0$, and gray line represents the maximum idf value (8.28))

4. RESULTS

4.1 Filtering hash occurrences according to tf-idf values

We can observe that the database response size decreases abruptly as the value of τ gets around the maximum idf value, after some oscillations with respect to a baseline, and finally resting in very low values for any $\tau > 15$, as seen in Figure 1. In this figure, each line indicates, for different query sizes, the amount of results and the average number of hashes per result given a value of τ. All values fall abruptly at about the same position. An attempt to explain the reason for this behavior can be found at the end of this section.

As expected, the accuracy of the system is also affected by increasing the τ parameter value. The Figure 2 shows that the accuracy follows a similar behavior as compared to the payload size. However, only values close to the maximum accuracy interest us, which leads us to consider what would be the best τ where accuracy is maximal and the response size is minimal. Figure 3 combines these two aspects within the same view: the blue line represents the accuracy while the orange line represents the payload size as long as queries are filtered by τ. The red line represents the chosen τ, while the gray one indicates the maximum idf in the database.

4.2 Evaluating cosine similarity and other methods as alternative scoring strategy

The outcomes show that cosine similarity measure can indeed be used as an alternative to the original measure. In a scenario where $\tau = 0$ (where amount of results and payload size are not filtered), the COS scoring method was shown to preserve accuracy (as compared to WNG). Table 1 shows the overall results for $\hat{\tau}$ on the three datasets used. It is noticeable that in most of the cases the accuracy obtained with Wang's original alignment method [2]

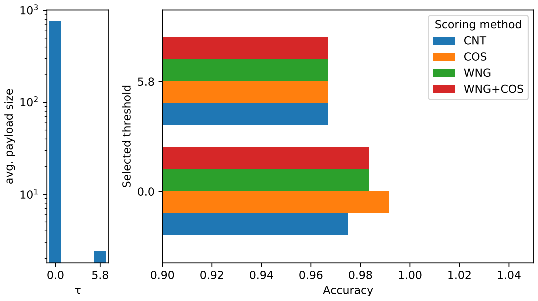

Figure 4. Results comparison for GTZAN dataset

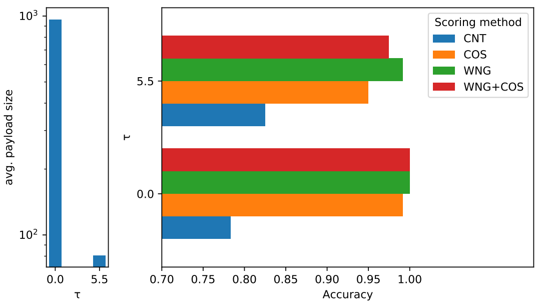

Figure 5. Results comparison for YTC+ dataset

is preserved when using the cosine similarity. In the YTC+ dataset, the original WNG alignment still outperforms the other methods (see also Figure 5), but we argue that this small loss of accuracy might be worth it, since the original method compromises the algorithm's robustness against time distortions. Still, a lower, more conservative value of τ could also be selected, granting best accuracy values for a small price in the payload size.

Another interesting observation is that the combined method WNG+COS didn't provided good results on any dataset, with performance always between COS and WNG but still preserving WNG's temporal restrictions. As expected, the CNT method had the worst performance in all scenarios.

Figures 4 and 5 provide a comparison of the results for different datasets. Notice that in both the normalized average amount of results (orange) and the normalized average payload size (green), all attained values are a fraction of the corresponding values of Shazam's original algorithm ($\tau = 0$ and scoring method WNG).

4.3 Investigating the sudden decrease of response sizes

Additional analysis was conducted in order to better understand the sudden decrease of response size when τ exceeds a certain value. The Figure 6 represents the rank-frequency distribution of the indexed fingerprints. In a classical natural language text-retrieval scenario, we would expect to see a Zipfian distribution in this graphic, where the relation of frequency between two adjacent ranking positions correspond to a power law, and hence a linear behaviour would be seen in this log-log representation. However, the different nature of musical fingerprints show a very different behavior, with stable *plateaus* followed by steeper decreases. Figure 7, in turn, shows the frequency of fingerprints in the database grouped by their tf-idf values (blue dots) and the accumulated sum of these occurrences in the corpus (orange line). The initial *plateau* in the first figure suggests that less than one hundred fingerprints are accountable for the most expressive part of the corpus. By observing the second figure, we also notice that most fingerprints in the corpus present low tfidf values. It indicates that there is a relatively small set of very common, non-relevant fingerprints that are spread across the indexed songs. As long as we remove these elements from the query, many documents that share these less informative fingerprints are not considered as candidates for the query. The $\hat{\tau}$ chosen for this dataset (7.0) is marked with the red line, whereas about 88% of the fingerprints in the corpus are below this line. It is also noticeable that a very small amount of fingerprints in the database has tfidf values greater than $max(\mathrm{idf})$, which suggests that $\hat{\tau}$ should not exceed this value.

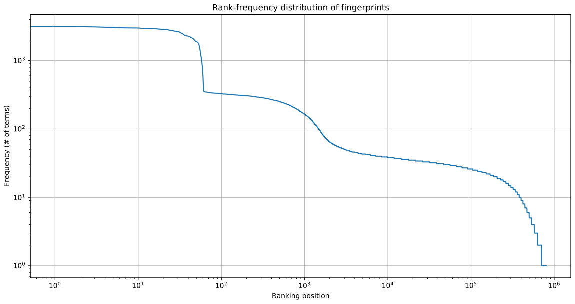

Figure 6. Rank-frequency distribution of hash occurrences

	GTZAN	FMA	YTC+
Total Records	961	7963	1534
$\hat{\tau}$	5.8	7	5.5
Accuracy (WNG, $\hat{\tau}$)	0.96	0.98	0.99
Accuracy (COS, $\hat{\tau}$)	0.96	0.98	0.95
Avg. payload size τ_0	759	6438	949
Avg. payload size $\hat{\tau}$	2	100	80
Payload size reduction	99.68%	98.45%	91.49%

Table 1. Overall results for different datasets

5. CONCLUSION

Filtering query fingerprints by tf-idf values is an effective approach to reduce the response's payload size with minor or null impact in the system's accuracy, and it brings many performance advantages to the system (network traffic, memory usage, database availability, etc). It is worth mentioning that many approaches could be used in order to select the best $\hat{\tau}$, and this choice depends on how tolerant the use-case is to loss of accuracy in exchange of a minimal payload size. In the special use case where the database will no longer receive new entries (i.e., it will only respond to queries), all the hash occurrences under the selected threshold value τ could be deleted, producing a considerable reduction of the database's size.

We conclude that cosine similarity can be used as an alternative of the original Shazam's scoring method, producing

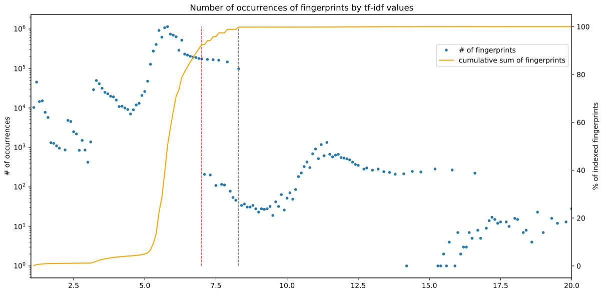

Figure 7. The blue line shows the log-scaled frequency of hash occurrences in the database such that $\mathrm{tfidf}_{h,d} = x$. The orange line shows the accumulated sum of hash occurrences at the same position. The red line indicates the selected t value for this instance, while the gray one shows the maximum idf value in the dataset.

better results when associated with tf-idf filtering ($\tau > 0$), and we also conclude that this approach is more suitable for projecting systems where robustness against time distortions is part of the requirements. With the exception of some extra calculations at the input phase (e.g., when inserting new audio clips) and the need of storing $\mathrm{tf}_{h,d}$ values, no other disadvantages of using such techniques were found. Automatically determining the optimal τ, specially in dynamic scenarios where new audio clips are constantly indexed over time, is a subject to be addressed in further work.

6. REFERENCES

[1] J. Haitsma and T. Kalker, “A Highly Robust Audio Fingerprinting System With an Efficient Search Strategy,” *Journal of New Music Research*, vol. 32, no. 2, pp. 211–221, 2003.

[2] A. L.-c. Wang and K. H. Street, “An Industrial-Strength Audio Search Algorithm,” in *International Conference on Music Information Retrieval ISMIR*, vol. 2, no. 2, 2003, pp. 7–13.

[3] T. Bertin-mahieux, D. P. W. Ellis, S. W. Mudd, W. Street, and N. York, “Large-scale cover song recognition using hashed chroma landmarks, LabROSA , Columbia University,” *Waspaa*, pp. 10–13, 2011.

[4] M. Riley, E. Heinen, and J. Ghosh, “A text retrieval approach to content-based audio retrieval,” in *Int. Symp. on Music Information Retrieval (ISMIR)*, 2008, pp. 295–300.

[5] V. Rotteker, “Evaluation of Pitch Shift and Time Stretch Invariant Acoustic Fingerprint Systems,” Ph.D. dissertation, 2016.

[6] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[7] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[8] D. Lippman, *Math in Society*, 2012. [Online]. Available: http://www.opentextbookstore.com/mathinsociety/

[9] D. P. W. Ellis, “2014-DP1 - The 2014 labrosa audio fingerprint system,” pp. 6–7, 2014.

[10] G. Tzanetakis, G. Essl, and P. Cook, “Automatic Musical Genre Classification Of Audio Signals,” 2001.

[11] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A Dataset for Music Analysis,” in *18th International Society for Music Information Retrieval Conference*, 2017.

[12] D. F. Silva, V. M. A. Souza, and G. E. A. P. A. Batista, “Music Shapelets for Fast Cover Song Recognition,” in *ISMIR*, 2015.