LANDMARK-BASED AUDIO FINGERPRINTING FOR DJ MIX MONITORING

Reinhard Sonnleitner[1], **Andreas Arzt**[1], **Gerhard Widmer**[1,2]
Department of Computational Perception, Johannes Kepler University, Linz, Austria[1]
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria[2]
reinhard.sonnleitner@jku.at

ABSTRACT

Recently, the media monitoring industry shows increased interest in applying automated audio identification systems for revenue distribution of DJ performances played in discotheques. DJ mixes incorporate a wide variety of signal modifications, e.g. pitch shifting, tempo modifications, cross-fading and beat-matching. These signal modifications are expected to be more severe than what is usually encountered in the monitoring of radio and TV broadcasts. The monitoring of DJ mixes presents a hard challenge for automated music identification systems, which need to be robust to various signal modifications while maintaining a high level of specificity to avoid false revenue assignment. In this work we assess the fitness of three landmark-based audio fingerprinting systems with different properties on real-world data – DJ mixes that were performed in discotheques. To enable the research community to evaluate systems on DJ mixes, we also create and publish a freely available, creative-commons licensed dataset of DJ mixes along with their reference tracks and song-border annotations. Experiments on these datasets reveal that a recent quad-based method achieves considerably higher performance on this task than the other methods.

1. INTRODUCTION

Automated audio identification systems, also referred to as audio fingerprinters, identify a piece of query audio from a collection of known reference audio pieces. In general, such systems search for characteristic features in the query audio, which are then compared to features of known audio pieces. The features are the so-called fingerprints, which should embody a favourable trade-off in storage demands, computation complexity, comparability, specificity, and robustness. The importance of the individual properties of the fingerprints is dictated by the use case. The industry uses audio identifications systems to monitor radio and TV broadcast channels to create detailed lists of the specific content that was played at any given time. In addition to radio and TV broadcast monitoring, performance rights organizations show interest in monitoring music performances, for example in discotheques. Without using automated identification systems, royalty collection depends on the broadcasters who are expected to create detailed lists of played content.

Musical content that is played in discotheques is usually performed by DJs, who can introduce severe signal modifications by mixing sets of songs in a homogeneous fashion. This frequently involves temporally changing the pitch or tempo of the audio to achieve a smooth transition from one track to the other, and often DJs will add effects in response to the mood or atmosphere in the club.

Signal content that is modified by DJs arguably puts enormous robustness demands on automated systems. It seems hard to quantify the type and severity of signal manipulations that can be introduced by DJs, as several effects can be applied in combination. For the same reason we believe it is hard to manually create meaningful test cases that reflect the possible modifications for system evaluation.

In this work, we investigate the fitness and performance of systems that belong to the class of so-called *landmark-based* audio fingerprinting methods. Landmark-based systems extract highly robust feature points, i.e. local energy maxima, from the two dimensional time-frequency representation of the audio signal, and combine groups of these landmarks to form the individual fingerprints.

We show via experiments that it is hard to achieve accurate results on DJ mixes. To do this, we test three implementations with different robustness properties, and report on their abilities to correctly identify known audio pieces while correctly abstaining from reporting a match if the correct song is not contained in the given reference database.

While the algorithmic approaches that we use in this work are extensively evaluated in the literature, we show that the application to DJ mixes indeed unveils shortcomings, specifically in the ability to prevent false detections. In the context of media monitoring, falsely detecting a song can lead to incorrect royalty management.

We contribute a new dataset which poses difficulties to automated identification systems, and investigate the different properties of three landmark based systems via experiments on these datasets.

The paper is organized as follows. Section 2 discusses prior and related work, in Section 3 we introduce the

datasets that are the basis for the experiments and analysis and interpretation of results. Section 4 gives an overview of the methods we test in this work. Then, in Section 5 we describe the setup of experiments and their evaluation. An analysis of the different properties of the tested methods is given in Section 6. Finally, in Section 7 we conclude our work.

2. RELATED WORK

The field of audio fingerprinting enjoys high research activity and numerous systems are described in the literature that approach the task [6, 7, 10, 12, 14, 15, 18]. Excellent reviews of earlier systems are presented in [2, 3].

The system described in [13] achieves pitch scale change robustness to small scaling factors by describing content based on short term band energies. In addition, the system is robust to small time scale modifications.

The basic algorithm of the Shazam system, a well known representative method for landmark-based audio fingerprinting, is described in [18]. It pairs spectral peaks (i.e. the so-called landmarks) that are extracted from the audio to obtain compact hashes, which are used as index into a hashtable to search for matches. The fingerprints are highly robust to environmental noise and signal degradations that result from digital-analog conversions of audio.

Another system that achieves a certain amount of robustness to changes in playback speed is described in [4]. As the change of the playback speed of audio influences the pitch scale, a system is described that can mitigate this effect by first searching for common pitch offsets of query and reference pieces, and then rescaling the query accordingly. This system also is a member of landmark based identification methods.

The work described in [1, 19, 20] incorporates techniques from the domain of computer vision to audio identification. The authors of [1] apply a wavelet transform to signals and create compact bit vector descriptors of content that can be efficiently indexed via the Min-Hash algorithm. The approach shown in [20] uses the image retrieval and similarity approach by applying the SIFT [9] method on logarithmically scaled audio spectrograms, and later propose a matching method using LSH [5] in [19].

The concept of extracting features based on time-chroma patches from the time frequency representation of the audio to describe robust features for audio identification is discussed in [11].

We proposed to perform audio identification using compact scale invariant quad descriptors that are robust to time, pitch or speed modifications in [16], and later refined and extended that approach in [17].

The systems we use for the experiments in this work are described in Section 4.

3. DATA SETS

We perform experiments on two different datasets, called disco set, and mixotic set. In the following we introduce these datasets, and summarize their properties in Table 1.

Disco	tracks	ref.	$+[s]$	$-[s]$
set0	25	18	5661	2179
set1	12	12	3760	0
set2	12	11	3206	294
set3	11	4	1054	2006
set20	19	17	3123	457
set35	20	7	324	996
set36	28	13	872	768
set37	21	10	720	720
total: 8	148	92	18 720	7420
Mixotic	tracks	ref.	$+[s]$	$-[s]$
set044	14	14	4640	0
set123	12	12	3320	0
set222	18	11	3543	2097
set230	9	7	2560	780
set275	17	11	3398	1622
set278	12	11	3576	284
set281	18	15	3300	280
set282	14	8	2200	1740
set285	15	15	4540	0
set286	14	14	3140	0
total: 10	143	118	34 217	6803

Table 1: Data set properties of the disco set (top) and the mixotic set (bottom). The column "tracks" gives the number of played tracks in the DJ mix, "ref" denotes the number of these tracks that are present in the reference database, and the columns "$+[s]$, $-[s]$" hold the number of seconds of referenced audio and not-referenced audio for the individual DJ mixes.

The first dataset, the *disco set*, contains eight mixes that were performed in discotheques, and digitally recorded from the DJ mixing desk. The duration of the mixes is approximately 7 hours and 16 minutes. For this dataset we have 296 reference tracks, only some of which are actually played in the mixes. The genres of the mixes include pop and rock, electronic music and German folk.

Because of copyright reasons, we cannot make the disco set publicly available, therefore we compile a second dataset, called *mixotic set*. We created this dataset from free, CC-licensed DJ mixes that were published on the mixotic netlabel [1] , and collected their respective reference songs, which are available under the same license. The mixotic set consists of 10 mixes with a total duration of 11 hours and 23 minutes. For this dataset we collected a set of 723 reference tracks, 118 of which are actually played in the mixes. According to the artists, this set contains genres like Techno, Chicago House, Deep-Tech, Dub-Techno, Tech-House, and the like. To be able to evaluate the fingerprinting results, we annotated the song borders of the tracks that are played in the individual mixes. Due to the long fading regions and sometimes very homogeneous track transitions, these annotations cannot be exact. We tried to mark the positions in time where the

[1] Mixotic is accessible via http://www.mixotic.net.

previous track is fully faded out.

We think that the mixotic set may be useful to the research community, and could help to design well balanced identification systems and to uncover specific strengths and potential shortcomings of various methods, therefore we publish the mixotic set along with the annotations [2].

4. METHOD OVERVIEW

We use the datasets that we described in the previous section to experiment with the following three methods: *Audfprint*, *Panako* and the quad based audio fingerprinter, henceforth referred to as simply *Qfp*.

Audfprint Audfprint is a MIT-licensed implementation [3] of a landmark-based audio identification algorithm based on the method described in [18]. The published algorithm utilizes quantized hash fingerprints that represent pairs of spectral peaks. The hashes are described by the time-frequency position of the first peak and its distance in time and frequency to the second peak. The hashes that are computed from a snippet of query audio are used as the keys into a suitable reference data structure, e.g. a hash table, to retrieve reference hashes with the same key. For each query hash, a lookup is performed and the result sets are collected. Matched query and reference hashes which happen to have a constant time offset in their individual peak-time identify the reference audio, along with its position in which the query snippet could be located.

Panako Panako [15], available [4] under the GNU Affero General Public License is a free audio identification system. It transforms the time domain audio signal into a two dimensional time frequency representation using the Constant Q transform, from which it extracts event coordinates. Instead of peak pairs, the method uses triples, which allows for a hash representation that is robust to small time and pitch scale modifications of the query audio. Thus, the system can also report the scale change factors of the query audio with respect to the identified reference. The system is evaluated on queries against a database of 30 000 full length songs, and on this data set achieves perfect specificity while being able to detect queries that were changed in time or frequency scale of up to around 8%. In this work we use Version 1.4 of Panako.

Qfp The Qfp method [16, 17] is a landmark based method that is robust to time and pitch scale changes of query audio. Its evaluation shows high average accuracy of more than 95% and an average precision of 99% on queries that are modified in pitch and/or time scale by up to $\pm 30\%$. The evaluation is performed on a reference data base consisting of 100 000 full length songs. The average query run time is under two seconds for query snippets of 20 seconds in length. The system also correctly uncovers any underlying scale changes of query audio. While some robust audio identification systems are using methods from the field of computer vision (c.f. Section 2), Qfp is inspired by a method used in astronomy [8], which proposes to use n-tuples (with $n > 2$) of two dimensional point coordinates to describe continuous feature descriptors that are invariant to rotation and *isotropic* scaling. The Qfp method adapts the described findings to represent *non-isotropic*-scale invariant features that allow for robust and efficient audio identification. The system uses range queries against a spatial data structure, and a subsequent verification stage to reliably discard false matches. The verification process accepts matches within individual match sequences if spectral peaks in a region around the candidate match in the reference audio are also present in the query audio excerpt. Evaluation results of the Qfp method along with a parameter study and resulting run times are given in [17].

These methods are well performing identification systems. An evaluation of experiments using Audfprint and Panako is given in [15]. While all three methods are landmark-based, the systems employ different inner mechanisms and thus are expected to perform differently on the datasets used in this work. Note that we use Audfprint and Panako as published, without tuning to the task at hand. We do this because we believe that the methods are published with a set of standard parameters that turned out to be well suited for general use cases according to experimentation performed by their authors. Likewise, we also use the same set of parameters for Qfp, as they are described in [17]. We incorporated improvements for runtime, but these do not have any impact on the identification results at all. For the task at hand, we want to investigate the fitness of the underlying algorithms of the methods, rather than discussing their specific implementations.

5. EXPERIMENT SETUP

Experiments are performed individually on the datasets we described in Section 3. The general experimental setup is as follows. The mixes are split into non-overlapping query snippets of 20 seconds in length. To create query snippets from the DJ mix we use the tool SoX [5] along with switches to prevent clipping, and convert the snippets into .wav files.

The methods process each query snippet and store the results. The implementations of the three tested systems behave differently in answering a query: if the query excerpt could be matched, Audfprint and Panako by default report the whole query duration as matched sequence. Qfp gives a more detailed answer and reports the start time and end time of the matched portion within the query excerpt. Likewise, as Qfp, Audfprint allows to report the exact part of the query that it could actually match (using the option `--find-time-range`), but for Panako we did not find such an option. For best comparability of the evaluation results, for all of the three methods we assign the reported match file ID to its whole query of 20 seconds.

[2] Available on http://www.cp.jku.at/datasets/fingerprinting/

[3] Audfprint is available on https://github.com/dpwe/audfprint.

[4] Panako is available on http://www.panako.be/.

[5] SoX is available on http://sox.sourceforge.net/.

Dataset				Referenced					not ref.		
	$+[s]$	$-[s]$	M.	TP	FP	FN	**acc.**	**prec.**	TN	FP	**spec.**
Disco.	18720	7420	**A**	7838	7440	3442	0.419	0.513	3611	3809	0.487
			P	4624	5596	8500	0.247	0.452	5539	1881	0.746
			Q	13879	1253	3588	**0.741**	**0.917**	6996	424	**0.942**
			Q^{v}	14316	1523	2881	0.765	0.904	6587	833	0.888
			Q^{ϵ}	3423	152	15145	0.183	0.957	7413	7	0.999
Mixotic	34 217	6803	**A**	21783	10233	2201	0.637	0.680	1735	5068	0.255
			P	12326	16181	5710	0.360	0.432	2371	4432	0.349
			Q	29985	1262	2970	**0.876**	**0.959**	6304	499	**0.927**
			Q^{v}	30445	1680	2092	0.889	0.948	4395	2408	0.647
			Q^{ϵ}	19497	349	14371	0.570	0.982	6715	88	0.987

Table 2: Evaluation results for the data sets. The column "+" shows the number of seconds of the DJ mix, for which a reference is present. The column "−" likewise gives the number of seconds for which no reference track is present in the database. The methods (M.) Audfprint, Panako and Qfp are abbreviated as "A", "P" and "Q". The column "Q^{v}" shows Qfp results without the verification stage, and "Q^{ϵ}" shows the results for reduced search neighbourhood. "acc." is the accuracy, "prec." is the precision and "spec." is the specificity. The experiment setup and the meaning of the measures is defined in Section 5. Because of space constraints we omit showing the individual statistics of each DJ mix that is contained in the dataset, and directly present the overall values. Detailed results are available in the published dataset.

It is important to note that we do not perform smoothing over time on the individual results but rather test the *raw* identification performance of each method based on each individual query.

We compare the fingerprinting results to the ground truth on a one second basis, i.e. for each second of the DJ mix we check whether the corresponding query result is correct.

Here we distinguish the following two cases: Case 1 $(C1)$ *identifiable*, and Case 2 $(C2)$ *not identifiable* portions of the mixes. We investigate how the systems perform in cases where a song is identifiable, because it is present in the reference database $(C1)$, and how well behaving a system is in not producing a match result in cases where this is correct, i.e. because the track is in fact not present in the reference $(C2)$.

For all cases $(C1)$, we count the number of seconds of true positives (TP), false positives (FP) and false negatives (FN). True positives are cases in which the system correctly identified a track from a query. The false positives denote situations in which the wrong track is claimed to be present, and the false negatives are cases in which the system did not report a result at all. For this evaluation there exist no true negatives, i.e. $TP+FP+FN = N$. For this case $(C1)$ we define the following two performance measures.

Accuracy, as the proportion of correctly identified queries:

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{TP}{N} \tag{1}$$

Precision, as the proportion of cases in which the system reports an identification and this claim is correct, i.e. a system that operates with high precision produces a low proportion of false positives:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

To assess system performance for cases $(C2)$, in which the reference track is unknown, i.e. not present in the database, we compute a third evaluation measure, the specificity:

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

Here, TN denotes the number of seconds in which no result was produced, and at the same time the reference track is absent. The number of FP are the cases where the system reports a match despite the fact that there is no reference. Specificity expresses the capability of a system to avoid false predictions by not reporting a result.

The identification performance of all three methods is listed in Table 2. We will discuss the results in the Section below, and analyze the properties and differences of the methods.

6. DISCUSSION OF RESULTS

Table 2 summarizes the results of each method on the disco set and the mixotic set (rows Q^{v} and Q^{ϵ} become relevant at a later point of this section). For the disco set, the accuracy shows that just between 25% and 74% of detectable seconds were assigned to the correct reference track. This reveals that DJ mix track identification indeed is a tough problem. The precision values show that Audfprint and Panako claim a wrong track in around 50% of the cases where the correct track should be identifiable. The specificity of the systems shows that Audfprint correctly abstains from claiming a match in roughly 50% of the cases where no track can be found because it is not referenced in the database. Panako shows higher specificity at around 75%. Qfp manages to correctly treat TN in 94% of the cases.

The results obtained from the experiment on the mixotic set show better accuracy for all three methods, and Audfprint and Qfp operate with higher precision than on the disco set. For the mixotic set, all three systems show lower

specificity than for the disco set. We believe that this is a result of the larger reference database (723 songs rather than 296 in the disco set) and the highly repetitive tracks in the mixotic set. In total, Qfp performs at higher accuracy, precision and specificity than Audfprint and Panako. Panako shows higher specificity than Audfprint on both datasets.

The low specificity of the algorithm that is implemented in Audfprint indicates that its fingerprints are too general. Panako uses triples of peaks, which inherently capture more specific information of the local signal. Indeed, its specificity on the disco set is considerably higher than that of Audfprint, i.e. its fingerprint descriptors are less general, which may be the reason for it to correctly refuse to make a claim in around 75% of the cases on the disco set, and in roughly 35% of the cases on the mixotic set.

Analysis Qfp performs best on the tested datasets. To find out which properties of the system are responsible for that, we perform two additional experiments. The first experiment is intended to investigate the impact of the verification process, and the second experiment highlights the effect of the range query for Qfp. For a detailed explanation on the parameters that are mentioned in this section, we ask the reader to consult [17].

First, we want to find out if it is the verification process that allows it to maintain high performance.

If we switch off the verification [6] and run the experiments, this results in an overall accuracy of 0.76, a precision of 0.90, and a specificity of 0.89 on the disco set. For the mixotic set this results in the accuracy of 0.89, precision of 0.95 and a specificity of 0.65 (c.f. Table 2, row Q^v). In terms of accuracy and precision, the results for both datasets are comparable to those with active verification. The specificity on the mixotic set, however, is notably lower.

We now investigate the performance of the Qfp method using a reduced neighbourhood for the range queries. We argue, that this loosely translates to using quantized hashes, i.e. if a query peak moves with respect to the others, the corresponding reference hash cannot be retrieved. This neighbourhood is specified as distance in the continuous hash space of the quad descriptor. For this experiment we reduce this distance from $0.0035, 0.012$ for pitch and time to $0.001, 0.001$ for pitch and time. For the disco set, this results in a low accuracy of 0.18, precision of 0.96 and specificity of 0.99. On the mixotic set, the small range query neighbourhoods result in an accuracy and precision of 0.57 and 0.98, and specificity of 0.99 (c.f. Table 2, Q^ϵ).

Extended Database We now add the reference tracks of both, the disco set and the mixotic set to a reference database that consists of $430\,000$ full length tracks (this captures almost the entire Jamendo corpus [7]), and inspect how the Qfp method responds to that amount of additional tracks. The overall result for the disco set (with standard settings for the range query and verification) is 0.69 for accuracy and 0.80 for precision. The specificity is 0.71. On the mixotic set, the results are as follows: Accuracy 0.83, precision 0.87 and specificity 0.56. The low specificity here is also impacted by a song duplicate in the DJ mixes and Jamendo corpus, i.e. in the case of mixotic set 282, Qfp could correctly identify the track "Akusmatic - Scamos" within the additional $430\,000$ songs, but the evaluation treats this as FP, because according to the ground truth this track is not present. The issue with song duplicates does not influence any other experiments in this work, since we use the extended reference database only with the Qfp method.

The experiment shows that there is a certain negative impact, causing more FP when trying to identify tracks in DJ mixes on larger databases. Note that these results also depend on the experiment setup as defined in Section 5, where we chose to assign the identified track ID to the whole query of 20 seconds in length. If we respect the reported start and end time of identified queries, the results on the disco set give an accuracy of 0.60, precision of 0.88, and a specificity of 0.89. For the mixotic set the accuracy then is 0.76, precision is 0.93 and the specificity results in 0.80.

Qfp turns out to maintain – what we think is – acceptable performance, on a database with $430\,000$ full length songs. According to precision and specificity, the other methods tested in this work seem to get distracted by 723 reference songs. This leads us to suggest that the monitoring of DJ mixes via automated fingerprinting systems indeed is a challenging task.

Visual analysis The different behaviour of the systems can be conveyed visually. In Figure 1 we show an excerpt of the *mixotic set* mix-ID 222 [8], from second 1500 to 4300. Vertical lines represent song borders. The figure shows the scattered query identification results, where the x-axis position is the query time, and the y-axis position locates the query within the reference song that the system could identify. Thus, scattered positions of songs that are correctly identified over several successive queries usually take the shape of a sawtooth function. In DJ mixes this will not always be the case, as the DJ can loop content. The different track names are encoded as markers, to be able to see if a system tends to confuse the same two tracks, or whether it reports many different tracks for a portion that it fails to identify correctly. The larger markers shown on top, between song borders, are the reference. A missing reference marker means that the song is not present in the database. Note that the evaluation does not consider whether the predicted position within the reference is correct, as this is not meaningful for highly repetitive musical content.

[6] Strictly speaking, the implementation does not allow to switch off the verification. Therefore we instead relax the verification constraints such that no candidate can be rejected.

[7] Jamendo is accessible via https://www.jamendo.com.

[8] The mix-IDs are listed and explained in the published dataset.

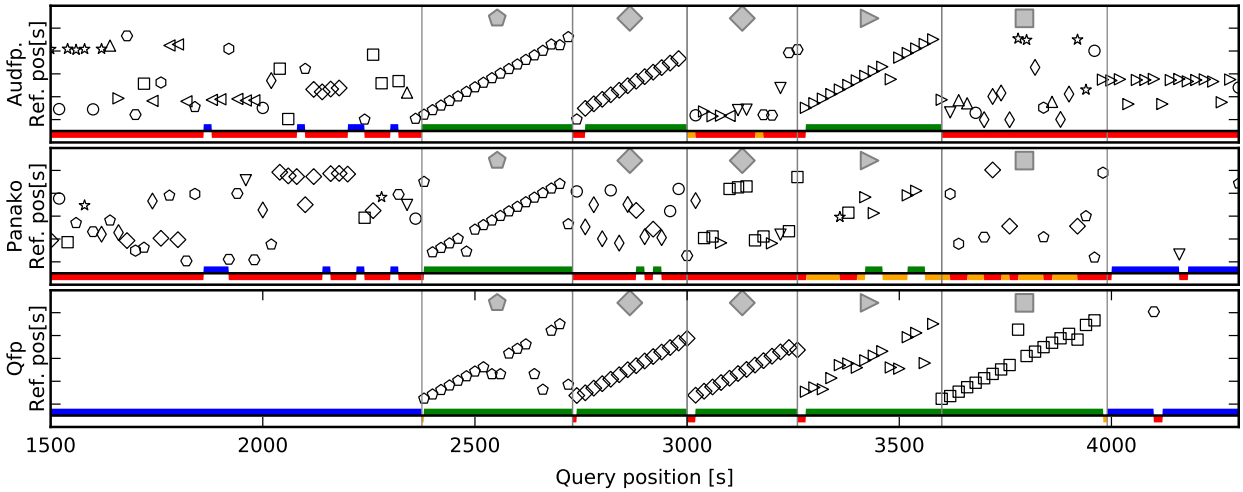

Figure 1: Query visualisation of an excerpt of mixotic set-ID 222. The rows show the results of individual, non-overlapping $20s$ queries without smoothing of predictions for Audfprint (top), Panako (middle) and Qfp (bottom). The vertical lines are the annotated song borders. The identification claims of the systems are encoded in the shown markers, where each marker represents a reference track. The x-axis position shows the query excerpt position, and y-axis the location of the matched query within the identified reference track. A missing large marker between song borders means that the reference song is not present in the database. The figures show a bar at the bottom, which represents the confusions. TP (green) and TN (blue) are shown on top of the horizontal line, FP (red) and FN (yellow) are shown below.

7. CONCLUSIONS

The results obtained from the experiments shown in this work support the intuition that automated audio identification on DJ mixes is a challenging problem. We observe that the Qfp method performs best on the tested datasets, and believe that it constitutes a well suited method to further investigate the analysis of DJ mixes via audio fingerprinting.

For future work and experiments we strive to collect DJ mixes with accurate annotations and timestamps, that are exported from the specific software or the midi controller used by the DJ. This would allow to gain insight on what kinds of effects and combinations thereof prevent automated identification systems from correctly identifying certain portions of query audio.

8. ACKNOWLEDGEMENTS

This work is supported by the Austrian Science Fund FWF under projects TRP307 and Z159, and by the European Research Council (ERC Grant Agreement 670035, project CON ESPRESSIONE).

9. REFERENCES

[1] Shumeet Baluja and Michele Covell. Audio fingerprinting: Combining computer vision & data stream processing. In *Acoustics, Speech and Signal Processing (ICASSP), 2007. IEEE International Conference on*, volume 2, pages II–213. IEEE, 2007.

[2] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of algorithms for audio fingerprinting. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 169–173. IEEE, 2002.

[3] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(3):271–284, 2005.

[4] Elsa Dupraz and Gaël Richard. Robust frequency-based audio fingerprinting. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 281–284. IEEE, 2010.

[5] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[6] Frank Kurth, Thorsten Gehrmann, and Meinard Müller. The cyclic beat spectrum: Tempo-related audio features for time-scale invariant audio identification. In *ISMIR*, pages 35–40, 2006.

[7] Frank Kurth, Andreas Ribbrock, and Michael Clausen. Identification of highly distorted audio material for querying large scale data bases. In *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[8] Dustin Lang, David W Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 137:1782–2800, 2010. arXiv:0910.2233.

[9] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[10] Chun-Shim Lu. Audio fingerprinting based on analyzing time-frequency localization of signals. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 174–177. IEEE, 2002.

[11] Mani Malekesmaeili and Rabab K Ward. A local fingerprinting approach for audio copy detection. *Signal Processing*, 98:308–321, 2014.

[12] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *ISMIR*, volume 2005, page 6th, 2005.

[13] Mathieu Ramona and Geoffroy Peeters. Audioprint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 818–822. IEEE, 2013.

[14] Joren Six and Olmo Cornelis. A robust audio fingerprinter based on pitch class histograms applications for ethnic music archives. In *Proceedings of the Folk Music Analysis conference (FMA 2012)*, 2012.

[15] Joren Six and Marc Leman. Panako - a scalable acoustic fingerprinting system handling time-scale and pitch modification. In *ISMIR*, pages 259–264, 2014.

[16] Reinhard Sonnleitner and Gerhard Widmer. Quad-based audio fingerprinting robust to time and frequency scaling. In *Proceedings of the 17th International Conference on Digital Audio Effects, DAFx-14, Erlangen, Germany, September 1-5, 2014*, pages 173–180, 2014.

[17] Reinhard Sonnleitner and Gerhard Widmer. Robust quad-based audio fingerprinting. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 24(3):409–421, 2016.

[18] Avery L Wang. An industrial-strength audio search algorithm. In *ISMIR*, pages 7–13, 2003.

[19] Xiu Zhang, Bilei Zhu, Linwei Li, Wei Li, Xiaoqiang Li, Wei Wang, Peizhong Lu, and Wenqiang Zhang. Sift-based local spectrogram image descriptor: a novel feature for robust music identification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):1–15, 2015.

[20] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 987–990. ACM, 2010.