A Survey on Music Retrieval Systems Using Microphone Input

Ladislav Maršík[1], Jaroslav Pokorný[1], and Martin Ilčík[2]

[1] Dept. of Software Engineering, Faculty of Mathematics and Physics
Charles University, Malostranské nám. 25, Prague, Czech Republic
{marsik, pokorny}@ksi.mff.cuni.cz
[2] The Institute of Computer Graphics and Algorithms,
Vienna University of Technology, Favoritenstraße 9-11, Vienna, Austria
1040 Vienna, Austria
ilcik@cg.tuwien.ac.at

Abstract. Interactive music retrieval systems using microphone input have become popular, with applications ranging from whistle queries to robust audio search engines capable of retrieving music from a short sample recorded in noisy environment. The availability for mobile devices brought them to millions of users. Underlying methods have promising results in the case that user provides a short recorded sample and seeks additional information about the piece. Now, the focus needs to be switched to areas where we are still unable to satisfy the user needs. Such a scenario can be the choice of a favorite music performance from the set of covers, or recordings of the same musical piece, e.g. in classical music. Various algorithms have been proposed for both basic retrieval and more advanced use cases. In this paper we provide a survey of the state-of-the-art methods for interactive music retrieval systems, from the perspective of specific user requirements.

Keywords: music information retrieval, music recognition, audio search engines, harmonic complexity, audio fingerprinting, cover song identification, whistling query

1 Introduction

Music recognition services have gained significant popularity and user bases in the recent years. Most of it came with the mobile devices, and the ease of using them as an input for various retrieval tasks. That has led to the creation of Shazam application[3] and their today's competitors, including SoundHound[4] or MusicID[5], which are all capable of retrieving music based on a recording made with a smartphone microphone. Offering these tools hand in hand with a convenient portal for listening experience, such as Last.fm[6] or Spotify[7], brings a

[3] http://www.shazam.com
[4] http://www.soundhound.com
[5] http://musicid2.com
[6] http://www.last.fm
[7] http://www.spotify.com

M. Nečaský, J. Pokorný, P. Moravec (Eds.): Dateso 2015, pp. 131–140, CEUR-WS.org/Vol-1343.

whole new way of entertainment to the users' portfolio. In the years to come, the user experience in these applications can be enhanced with the advances in music information retrieval research.

1.1 Recent Challenges in Music Retrieval

With each music retrieval system, a database of music has to be chosen to propel the search, and if possible, satisfy all the different queries. Even though databases with immense numbers of songs are used, such as the popular Million Song Dataset [1], they still can not satisfy the need to search music in various genres. At the time of writing of this paper, the front-runners in the field as Shazam Entertainment, Ltd., are working on incorporating more Classical or Jazz pieces into their dataset, since at the moment their algorithm is not expected to return results for these genres [22].

Let us now imagine a particular scenario – the user is attending dance classes and wishes his favorite music retrieval application to understand the rhythm of the music, and to output it as a result along with other information. Can the application adapt to this requirement?

Or, if the user wishes to compare different recordings of the same piece in Classical music? Can the resulting set comprise of all such recordings?

There are promising applications of high-level concepts such as music harmony to aid the retrieval tasks. De Haas et al. [2] have shown how traditional music theory can help the problem of extracting the chord progression. Khadkevich and Omologo [9] showed how the chord progression can lead us to an efficient cover identification. Our previous work [13] showed how music harmony can eventually cluster the data by different music periods. These are just some examples of how the new approaches can solve almost any music-related task that the users can assign to the system.

1.2 Outline

In this work we provide a survey of the state-of-the-art methods for music retrieval using microphone input, characterized by the different user requirements. In Section 2 we describe the recent methods for retrieving music from a query created by sample song recording using a smartphone microphone. In Section 3 we show the methods for the complementary inputs such as humming or whistling. We also look at the recent advances in cover song identification, in Section 4. Finally, we form our proposals to improve the recent methods, in Section 5.

2 Audio Fingerprinting

We start our survey on music retrieval systems with the most popular use case – queries made by recording a playback sample from the microphone and looking for an exact match. This task is known in music retrieval as *audio fingerprinting*. Popularized by the Shazam application, it became a competitive field in both academic and commercial research, in the recent years.

2.1 Basic Principle of Operation

Patented in 2002 by Wang and Smith [21], the Shazam algorithm has a massive use not only because of the commercial deployment, but mainly due to its robustness in noisy conditions and its speed. Wang describes the algorithm as a „combinatorially hashed time-frequency constellation analysis“ of the audio [22]. This means reducing the search for a sound sample in the database to a search for a graphical pattern.

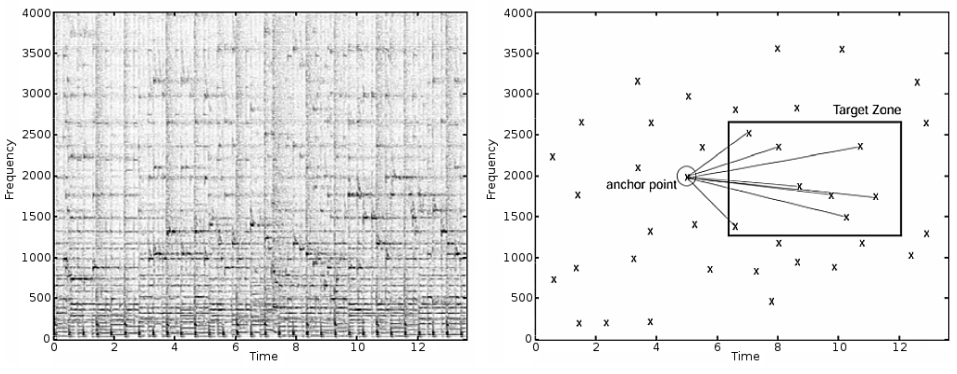

Fig. 1. On the left, time-frequency spectrogram of the audio, on the right, frequency peaks constellation and combinatorial hash generation. Image by Wang and Smith [22].

First, using a discrete-time Fourier transform, the time-frequency spectrogram is created from the sample, as seen on Figure 1 on the left. Points where the frequency is present in the given time are marked darker, and the brightness denotes the intensity of that particular frequency. A point with the intensity considerably higher than any of its neighbors is marked as a peak. Only the peaks stay selected while all the remaining information is discarded, resulting in a *constellation* as depicted on Figure 1 on the right. This technique is also used in a pre-processing step to extract the constellation for each musical piece in the database.

The next step is to search for the given sample constellation in the space of all database constellations using a pattern matching method. Within a single musical piece, it is the same as if we would match a small transparent foil with dots to the constellation surface. However, in order to find all possible matches, a large number of database entries must be searched. In our analogy, the transparent foil has the „width“ of several seconds, whereas the width of the surface constellation is several billion seconds, when summed up all pieces together. Therefore, optimization in form of combinatorial hashing is necessary to scale even to large databases.

As seen on Figure 1 on the right, a number of chosen peaks is associated with an „anchor“ peak by using a combinatorial hash function. The motivation behind using the fingerprints is to reduce the information necessary for search. Given the frequency f_1 and time t_1 of the anchor peak, the frequency f_2 and time t_2 of the peak, and a hash function h, the fingerprint is produced in the form:

$$h(f_1, f_2, t_2 - t_1)|t_1$$

where the operator | is a simple concatenation of strings. The concatenation of t_1 is done in order to simplify the search and help with later processing, since it is the offset from the beginning of the piece. Sorting fingerprints in the database, and comparing them instead of the original peak information results in a vast increase in search speed. To finally find the sample using the fingerprint matching, regression techniques can be used. Even simpler heuristics can be employed, since the problem can be reduced to finding points that form a linear correspondence between the sample and the song points in time.

2.2 Summary of Audio Fingerprinting and Benchmarking

Similar techniques have been used by other authors including Haitsma and Kalker [6] or Yang [23]. The approach that Yang uses is comparison of indexed peak sequences using Euclidean distance, and then returning a sorted list of matches. His work effectively shows how exact match has the highest retrieval accuracy, while using covers as input result in about 20% decrease in accuracy. As mentioned earlier, there are many other search engines besides Shazam application, each using its own fingerprinting algorithm. We forward the reader to a survey by Nanopoulos et al. [14] for an exhaustive list of such services.

To summarize the audio fingeprinting techniques, we need to highlight three points:

1. Search time is short, 5-500 milliseconds per query, according to Wang.
2. Algorithms behave greatly in the noisy environment, due to the fact that the peaks remain the same also in the degraded audio.
3. Although it is not the purpose of this use case, an improved version of the search algorithms could abstract from other characteristics, such as the tonal information (tones shifted up or down without affecting the result, we suggest Schönberg [16] for more information about tonality). However, the algorithms depend on the sample and the match being exactly the same in most of the characteristics, including tempo.

In the end, the algorithms are efficient in the use case they are devoted to, but are not expected to give results other than the exact match of the sample, with respect to the noise degradation.

Interestingly enough, a benchmark dataset and evaluation devoted to audio fingerprinting has only commenced recently[8], although the technology has been around for years. We attribute this to the fact that most of the applications were developed commercially.

[8] http://www.music-ir.org/mirex/wiki/2014:Audio_Fingerprinting

2.3 New Use Cases in Audio Search

There are other innovative fields emerging, when it comes to audio search. Notable are: finding more information about a TV program or advert, or recommendation of similar music for listening. Popularized first by the Mufin internet radio[9] and described by Schonfuss [17], these types of applications may soon become well-known on the application market.

3 Whistling and Humming Queries

Interesting applications arose with the introduction of „whistling“ or „humming“ queries. In this scenario, the user does not have access to the performance recording, but remembers the melody of the music she wants to retrieve. The input is whistling or humming the melody into the smartphone microphone.

3.1 Basic Principle of Operation

In their inspiring work, Shen and Lee [18] have described, how easy it is to translate a whistle input into MIDI format. In MIDI, musical sound commencing and halting are the events being recorded. Therefore, it is easily attainable from human whistle due to its nature. Shen and Lee further describe, that whistling is more suitable for input than humming, with the capture being more noise-resistant. Whistling has a frequency ranging from 700Hz to 2.8kHz, whereas other sounds fall under much smaller frequency span. String matching heuristics are then used for segmented MIDI data, featuring a modification of the popular *grep* Unix command-line tool, capable of searching for regular expressions, with some deviations allowed. Heuristics exist also for extracting melody from the song, and so the underlying database can be created from real recordings instead of MIDI. The whole process is explained in a diagram on Figure 2.

The search for the song in the database can be, as well as in Section 2, improved by forming a fingerprint and creating an index. Unal et al. [20] have formed the fingerprint from the relative pitch movements in the melody extracted from humming, thus increasing the certainty of the algorithm results.

3.2 Benchmarking for Whistling and Humming Queries

Many algorithms are proposed every year for whistling and humming queries. There is a natural need in finding the one that performs the best. The evaluation of the state-of-the-art methods can be found on annual benchmarking challenges such as MIREX[10] (Music Information Retrieval Evaluation Exchange, see Downie at al. [3] for details). The best performing algorithm for 2014 was the one from Hou et al. [8]. The authors have used Hierarchical K-means Tree (HKM) to enhance the speed and dynamic programming to compute the minimum edit

[9] http://www.mufin.com
[10] http://www.music-ir.org/mirex/wiki/MIREX_HOME

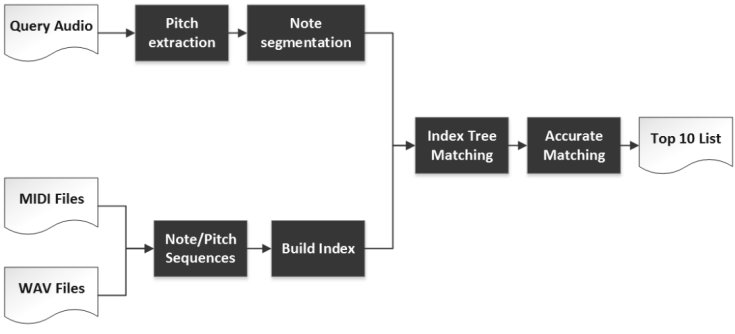

Fig. 2. Diagram of Query by Humming/Singing System, by Hou et al. [8].

distance between the note sequences. Another algorithm that outperformed the competition in the past years, while also being commercially deployed was MusicRadar[11].

Overall, whistling or humming queries are another efficient way of music retrieval, having already a number of popular applications.

4 Cover Song Identification Methods

In the last years, focus has switched to more specific use cases such as efficient search for the cover song or choosing from the set of similar performances. As described earlier, the exact-match result is not satisfying if we, for example, search for the best performance of Tchaikovsky's ballet, from a vast number of performances made. Although not geared on a microphone input (we are not aware of applications for such use case), this section provides an overview of recent cover song identification methods.

4.1 Methods Based on Music Harmony

The task requires a use of high-level concepts. Incorporation of music theory gives us the tool to analyze the music deeper, and find similarities in its structure from a higher perspective. The recent work of Khadkevich and Omologo [9] summarizes the process and shows one way how we can efficiently analyze the music to obtain all covers as the query result. The main idea is segmenting music to chords (musical elements in which several tones are sounding together). The music theory, as described e.g. by Schönberg [16] provides us with the taxonomy of chords, as well as the rules to translate between chords. Taking this approach, Khadkevich and Omologo have extracted „chord progression" data from a musical piece, and used Levenshtein's edit distance [11] to find similarities between

[11] http://www.doreso.com

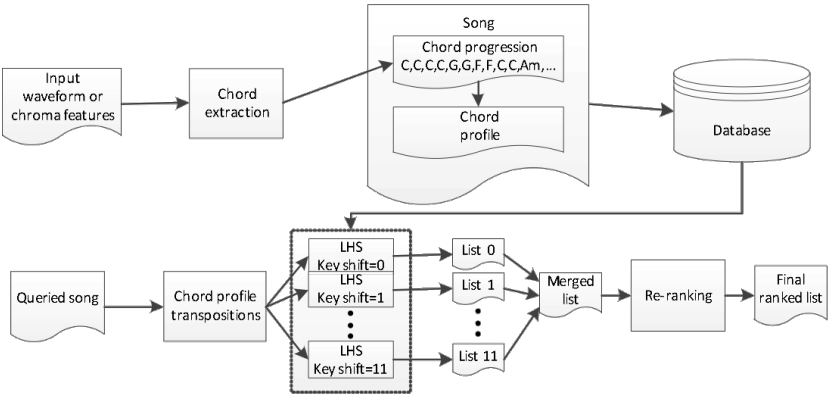

Fig. 3. Diagram of cover song identification by Khadkevich and Omologo [9].

the progressions, as depicted in Figure 3. A method of locality sensitive hashing was used to speed up the process, since the resulting progressions are high dimensional [5].

Another method was previously used by Kim et al. [10] at the University of Southern California. The difference between the approaches lay in the choice of fingerprints. Kim et al. have used a simple covariance matrix to mark down the co-sounding tones in each point of the time. Use of such fingerprints has, as well, improved the overall speed (approximately 40% search speed improvement over conventional systems using cross-correlation of data without the use of fingerprints). In this case, the fingerprints also improved the accuracy of the algorithm, since they are constructed in the way that respect music harmony. They also made the algorithm robust to variations which we need to abstract from, e.g. tempo. This can be attributed to the use of beat synchronization, described by Ellis and Poliner [4].

4.2 Benchmarking for Cover Song Identification

Same as in Section 3, cover song identification is another benchmarking category on annual MIREX challenge, with around 3-5 algorithms submitted every year. The best performing algorithm in the past few years was from The Academia Sinica and the team around Hsin-Ming Wang, that favored the use of extracting melody from song and using melody similarity [19]. Previous algorithm that outperformed the competition was the one made by Simbals[12] team from Bordeaux. The authors used techniques based on local alignment of chroma sequences (see Hanna et al. [7]), and have also developed techniques capable of identifying plagiarism in music (see Robine et al. [15]). On certain datasets, the mentioned

[12] http://simbals.labri.fr

algorithms were able to perform with 80-90% precision of identifying the correct covers.

5 Proposals for Improving Music Retrieval Methods

We see a way of improvement in the methods mentioned earlier. Much more can be accomplished if we use some standardized high-level descriptors. If we conclude that low-level techniques can not give satisfying results, we are left with a number of high-level concepts, which are, according to music experts and theoreticians, able to describe the music in an exhaustive manner. Among these the most commonly used are: *Melody*, *Harmony*, *Tonality*, *Rhythm* and *Tempo*. For some of these elements, it is fairly easy to derive the measures (e.g. Tempo, using the peak analysis similar to the one described in Section 2). For others this can be a difficult task and there are no leads what is the best technique to use. As a consequence, the advantage of using all of these music elements is not implemented yet in recent applications.

In our previous work we have defined the descriptor of Harmonic complexity [13], and described the significance of such descriptors for music similarity. The aim was to characterize music harmony in specific time of its play. We have shown that aggregating these harmony values for the whole piece can improve music recognition [12]. The next step, and possible improvement can be comparing the time series of such descriptors in music. Rather than aggregated values we can compare the whole series in time and obtain more precise results. Heuristics such as dynamic time warping can be used easily for this task. We now analyze the method and its impact on music retrieval. As the future work, experiments will take place to prove the proposed method.

Also, we see the option of combining general methods for cover song identification described in Section 4, with the use case of short recorded audio sample from the microphone. One of the possible ways is abstracting from tonal information and other aspects, as described briefly in Section 2.2. Recent benchmarking challenges for cover song identification are focusing on analyzing the whole songs, rather than a short sample. We believe that a combination of methods described in previous sections can yield interesting results and applications.

6 Summary and Conclusion

We have provided a survey of recent music retrieval methods focusing on: retrieving music based on audio input from recorded music, whistling and humming queries, as well as cover song identification. We described how the algorithms are performing efficiently in their use cases, but we also see ways to improve with new requirements coming from the users.

In the future work we will focus on the use of high-level descriptors and we propose stabilizing these descriptors for music retrieval. We also propose combining the known methods, and focusing not only on the mainstream music, but analyzing other genres, such as Classical, Jazz or Latino music.

Acknowledgments. The study was supported by the Charles University in Prague, project GA UK No. 708314.

Bibliography

1. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The Million Song Dataset. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. ISMIR 2011 (2011)
2. De Haas, W.B., Magalhães, J.P., Wiering, F.: Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. In: *Proceedings of the 13th International Society for Music Information Retrieval Conference*. ISMIR 2012 (2012)
3. Downie, J.S., West, K., Ehmann, A.F., Vincent, E.: The 2005 Music Information retrieval Evaluation Exchange (MIREX 2005): Preliminary Overview. In: *Proceedings of the 6th International Conference on Music Information Retrieval*. ISMIR 2005 (2005)
4. Ellis, D.P.W., Poliner, G.E.: Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP 2007 (2007)
5. Gionis, A., Indyk, P., Motwani, R.: Similarity Search in High Dimensions via Hashing. In: *Proceedings of the 25th International Conference on Very Large Data Bases*. VLDB '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
6. Haitsma, J., Kalker, T.: A Highly Robust Audio Fingerprinting System. In: *Proceedings of the 3rd International Society for Music Information Retrieval Conference*. ISMIR 2002 (2002)
7. Hanna, P., Ferraro, P., Robine, M.: On Optimizing the Editing Algorithms for Evaluating Similarity Between Monophonic Musical Sequences. Journal of New Music Research 36(4) (2007)
8. Hou, Y., Wu, M., Xie, D., Liu, H.: MIREX2014: Query by Humming/Singing System. In: *Music Information Retrieval Evaluation eXchange*. MIREX 2014 (2014)
9. Khadkevich, M., Omologo, M.: Large-Scale Cover Song Identification Using Chord Profiles. In: *Proceedings of the 14th International Society for Music Information Retrieval Conference*. ISMIR 2013 (2013)
10. Kim, S., Unal, E., Narayanan, S.S.: Music Fingerprint Extraction for Classical Music Cover Song Identification. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. ICME 2008 (2008)
11. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics-Doklady 10/8 (1966)
12. Marsik, L., Pokorny, J., Ilcik, M.: Improving Music Classification Using Harmonic Complexity. In: *Proceedings of the 14th conference Information Technologies - Applications and Theory*. ITAT 2014, Institute of Computer Science, AS CR (2014)
13. Marsik, L., Pokorny, J., Ilcik, M.: Towards a Harmonic Complexity of Musical Pieces. In: *Proceedings of the 14th Annual International Workshop on Databases, Texts, Specifications and Objects (DATESO 2014). CEUR Workshop Proceedings*, vol. 1139. CEUR-WS.org (2014)
14. Nanopoulos, A., Rafailidis, D., Ruxanda, M.M., Manolopoulos, Y.: Music Search Engines: Specifications and Challenges. Information Processing and Management: an International Journal 45(3) (2009)

15. Robine, M., Hanna, P., Ferraro, P., Allali, J.: Adaptation of String Matching Algorithms for Identification of Near-Duplicate Music Documents. In: *Proceedings of the International SIGIR Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*. SIGIR-PAN 2007 (2007)
16. Schönberg, A.: Theory of Harmony. University of California Press, Los Angeles (1922)
17. Schönfuss, D.: Content-Based Music Discovery. In: *Exploring Music Contents*, *Lecture Notes in Computer Science*, vol. 6684. Springer (2011)
18. Shen, H.C., Lee, C.: Whistle for Music: Using Melody Transcription and Approximate String Matching for Content-Based Query over a MIDI Database. Multimedia Tools and Applications 35(3) (2007)
19. Tsai, W.H., Yu, H.M., Wang, H.M.: Using the Similarity of Main Melodies to Identify Cover Versions of Popular Songs for Music Document Retrieval. Journal of Information Science and Engineering 24(6) (2008)
20. Unal, E., Chew, E., Georgiou, P., Narayanan, S.S.: Challenging Uncertainty in Query by Humming Systems: A Fingerprinting Approach. IEEE Transactions on Audio, Speech, and Language Processing 16(2) (2008)
21. Wang, A.L., Smith, J.O.: Method for Search in an Audio Database. Patent (February 2002), WO 02/011123A2
22. Wang, A.L.: An Industrial-Strength Audio Search Algorithm. In: *Proceedings of the 4th International Society for Music Information Retrieval Conference*. ISMIR 2003 (2003)
23. Yang, C.: Macs: Music Audio Characteristic Sequence Indexing for Similarity Retrieval. In: *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*. WASPAA 2001 (2001)