Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230572

The comparison of algorithms of audio fingerprinting

Linjun Xia

The College of Liberal Arts, University of Minnesota, Minneapolis, Minnesota, 55414, United States

Xia00068@umn.edu

Abstract. With the increasing demand for song recognition on music platforms, the Audio fingerprint system is being used more and more frequently. How to improve the speed and accuracy of the Audio fingerprint system is a problem that is worth studying. Finding a better algorithm is crucial if people who use this system want to improve their efficiency of it. This paper, it is compared the accuracy and speed of extracting music information by the Multiple Hashing Algorithm and the Philips Robust Hashing Algorithm. The Multiple Hashing Algorithm can improve the accuracy of information extraction by increasing the sub-fingerprint bands, but with the increase of sub-fingerprint bands, the algorithm run time is much longer. Therefore, the Philips Robust Hashing Algorithm should be used when processing simple audio information to ensure the speed of the system. The Multiple Hashing Algorithm can improve the accuracy of information extraction by increasing the number of sub-fingerprint bands.

Keywords: Audio fingerprint, Philips Robust Hashing, Multiple Hashing.

1. Introduction

The Audio fingerprint is a widely used system today. Many music software uses the audio fingerprint system to complete the user requirement of finding the corresponding song by using the song fragment. With the increasingly fierce competition of various companies, major applications are seeking to obtain accurate audio information at the fastest speed even in the case of poor sound quality and noisy environment. Improving the accuracy and speed of listening to songs and recognizing songs can improve the user experience and allow the app to gain a better reputation. In this context, the information acquisition accuracy, information acquisition speed, and matching speed of information in the database of the Audio fingerprint system are very important. This paper will also evaluate the pros and cons of the Philips Robust Hashing Algorithm and the Multiple Hashing Algorithm in terms of the accuracy and speed of finding information and the speed of matching database information. Before comparing the advantages and disadvantages of the two algorithms, this article will first introduce the basic principles of the Audio fingerprint system and the basic principles of the two algorithms.

The audio fingerprint system is mainly composed of two parts, the first part is to extract the track data in the music segment, and the second part is to compare the extracted data with the data in the database to find the corresponding target song. There are mainly five key factors in the system. These five factors determine the accuracy and speed of the system. They are Robustness, Reliability, Fingerprint size, Granularity, Search speed, and scalability [1]. Robustness refers to the ability to accurately extract sound features from sound information. Whether the sound quality is compressed or

DOI: 10.54254/2755-2721/5/20230572

distorted, regardless of whether the environment is noisy or not. Only the accurate extraction of audio information can improve the accuracy of system operation. Reliability refers to the ability of the system to find the difference between two similar pieces of sound information. Two songs sometimes have partially similar sound information. The system needs to find the difference between two similar sound clips to improve the accuracy of the system operation. Fingerprint size refers to the size of the extracted sound information. If the extracted information is too large, time and space for storage and extraction will be wasted, which will also affect the efficiency of system operation. Granularity refers to how long the system needs to collect information during the playback of sound information to find the target song. The less time spent collecting sound information, the less time the system operates, which increases system efficiency. Search speed and scalability refer to the time and storage capacity required by the system to match the resulting information with the database information. The shorter the time required and the smaller the storage space, the higher the operating efficiency of the system. The Multiple Hashing Algorithm and the Philips Robust Hashing Algorithm mentioned in this study will mainly compare their advantages and disadvantages from the aspects of Robustness and Granularity.

Before introducing the two algorithms in this article, it is important to introduce the fast Fourier transform (FFT). The "Fast Fourier Transform" (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal [2]. Both algorithms will work with the FFT.

The Philips Robust Hash (PRH) algorithm is divided into two stages. The first step is fingerprint extraction, the second step is database searching. The audio information entered in fingerprint extraction will be divided into overlapping frames. The length of each frame is about 370mm. This is to facilitate the application of the FFT function. The frame is then put into the FFT function for operation. Keep each move 1/32 the length of the frame while moving the frame. This results in 33 non-overlapping logarithmically spaced subbands at the end of the FFT function run. All frame, sub fingerprints, or hash strings are then computed [3]. At runtime, each entry in the hash table produces a list of pointers to audio files. The information of different frames is marked in the list. The algorithm will find the corresponding data in the database by matching the same pointer to complete the audio matching. The Multiple Hashing Algorithm (MLH) uses discrete cosine transform (DCT). It handles frames the same as PRH. The discrete cosine transform (DCT) is then utilized when dealing with sub-fingerprints. This makes it possible for each sub-fingerprint to be treated separately and improves overall efficiency by generating more sub-fingerprints.

2. Applications of Audio Fingerprint System

The Audio fingerprint system has a very important application prospect. Many technologies use the Audio fingerprint system to collect and compare sound systems. This section will introduce several applications of the Audio fingerprint system.

2.1. Audience measurement

The Audio fingerprint system can be used to calculate the number of viewers. When playing a program, the system can build a database according to the audio of the playing program and collect the audio information of the audience watching the program [4]. The extracted audio information is compared with the program information in the database. In this way, the ratings of different programs in different periods can be counted. Whether it is for the operator's program sequence or the improvement of the program itself. Statistical ratings are essential. This is an important application of the Audio fingerprint system.

2.2. Broadcast monitoring

Broadcast monitoring used to be the biggest application of the Audio fingerprint system. Its main application is in two aspects. The first aspect is the monitoring of advertisements. TV advertising is a very important marketing tool. In the advertising contract, both parties will stipulate the playing time

DOI: 10.54254/2755-2721/5/20230572

and the number of advertisements. If the merchant wants to monitor whether the platform plays the advertisement according to the contract requirements, it will set special audio for the advertisement to be placed. After that, by comparing the video data extraction of the advertising platform with the audio data of the advertisement through the Audio fingerprint system, the playback volume and playback time of the advertisement on the platform can be found.

The second major application is the monitoring of copyrights on major platforms. Several monitoring stations exist whether it is an Internet platform, a TV platform, or a radio platform. The monitoring station will put the audio information of the copyright data into the database. Each copyright information has a special data tag. The monitoring station will detect the broadcast of programs on various platforms and compare the program information of the major platforms with the special copyright marks in the database. In this way, if the platform uses data that it does not own the copyright, it will generate abnormal marks. The testing station will impose corresponding penalties on the infringement of the platform.

2.3. Find Music Automatically

Most of the most popular ways to use the Audio fingerprint system these days come from music applications. Many music apps can find music in their library by listening to clips of music played by the user. These music applications will extract music information from their copyrights to generate audio databases. When the user plays a music clip, the application will enable the Audio fingerprint system audio capture algorithm to capture feature information. Then compare the information with the audio information in the database and output the song information that matches successfully. This requires the algorithm to have accurate information and the ability to quickly match the information.

3. Algorithms

To better meet the needs of various applications, the algorithm of the Audio fingerprint system should have the ability to accurately capture data and quickly compare data. The algorithms used at this stage are mainly Philips Robust Hashing Algorithm and Multiple Hashing Algorithm.

3.1. Philips robust hashing algorithm

The Philips Robust Hashing Algorithm (PRH) is an algorithm proposed by Haitsma, Kalker, and Oostveen in 2001. In the Audio fingerprint system, there is often a contradiction between extracting audio information size and audio robustness. The system needs to extract as little information as possible to ensure the speed of extracting information and the running speed of the algorithm. But the smaller information can lead to distorted audio. So how to balance the size of the extracted information is the problem that Philips Robust Hashing Algorithm focuses on. Ideally, the extracted information should be as small as possible without loss of discriminative power and robust against imperceptible distortions of the original signal [5]. This algorithm divides the audio into small pieces, extracts the audio energy spectrum in a small piece and compares it with the database information immediately, and extracts the audio information of the next small piece at the same time. Overlapping and interleaving, not only ensures the extraction of small information, but also ensures the running speed of the system, and ensures that each segment of audio has better robustness. Therefore, this algorithm is widely used.

The Philips method is a statistical model used by the Philips Robust Hashing Algorithm when applied to the system. This model mainly utilizes the equivalent rearrangement theory. We set Vector as $X = (x[1], \dots\dots, x[N])^T$。 Put X as the input signal into the hash model. X will be split before being processed by the system. These split X's become overlapping frames. We then set M to be the number of samples contained in individual frames. We then set Δ to be the number of samples that do not overlap each other from frame to frame. In this way, the M-length of the variable X_n at the nth frame can be calculated. The formula is as follows:

$$X_n \triangleq (x[n * \Delta + 1], \dots\dots, x[n * \Delta + M])^T, n = 0,1,2 \dots\dots \quad (1)$$

DOI: 10.54254/2755-2721/5/20230572

The result obtained by the formula will be weighted before the fast Fourier transform. The data will then be recalculated through the hash algorithm to generate the hash number required by the system. The operation diagram of the algorithm is as follows:

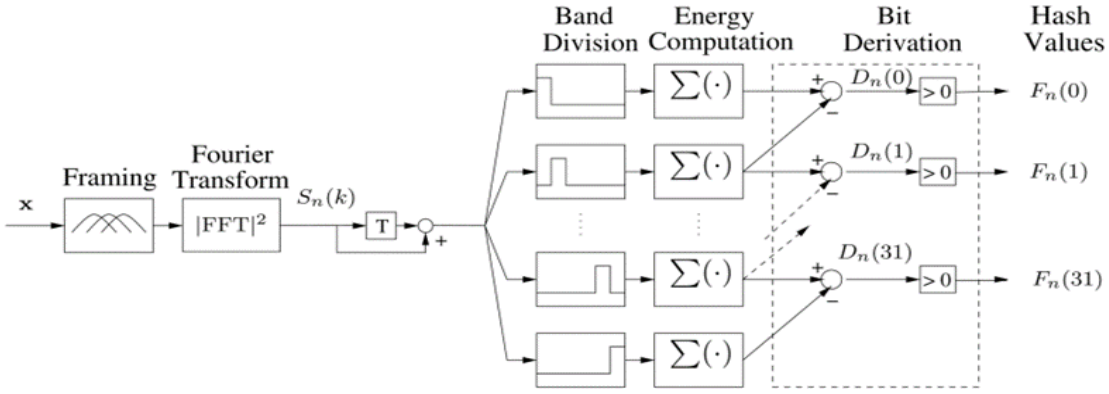

Figure 1. Philips Robust Hashing Algorithm how to work [6].

In addition to the outstanding performance of the Philips Robust Hashing Algorithm in information crawling, the algorithm also has unique features in database queries. The algorithm will generate a sub-fingerprint database from the collected audio information. Then use the list of each entry in the hash table to generate a pointer to the location of the sub-fingerprint. The position of the pointer is then compared with the data of the overall audio database. There will be a bit error rate threshold during comparison to ensure that some slightly distorted audio can accurately find the corresponding song. Usually, the bit error rate is set to 0.35.

3.2. Multiple hashing algorithm

The Multiple Hashing Algorithm (MLH) algorithm is a newer algorithm. The application of this algorithm in the audio fingerprint system is divided into two parts. The first part is the extraction of information. The second part is the comparison between extracted information and database information. MLH has a similar first half to PRH in extracting information. The MLH algorithm needs to first split the audio part into overlapping frames. The Fourier transform function is then used before the audio energy calculation. The sample energy segments of different frames are then calculated. But after that MLH uses the Discrete Cosine Transform (DCT) algorithm. The DCT algorithm is based on the fast Fourier transform. This algorithm can be used in the field of digital processing for pattern recognition and Wiener filtering [6]. The purpose of the MLH algorithm using DCT is to simplify the calculation and quickly determine the hash string. The MLH then utilizes the DCT in the temporal energy series. Framing different overlapping frames into subbands and calculating a DCT coefficient to generate sub-fingerprints. Only the lower-ordered K values from the DCT coefficients are kept for further computation of sub fingerprints [7]. Then the MLH algorithm will form database storage and construct multiple hash tables to calculate the sub fingerprints for each frame. The data extraction part ends with the construction of the hash table.

The database search in the second part consists of three steps. The first step divides the extracted audio into 256 frames and uses the hash table established before for operation [8]. In the second step, each hash table calculates a message that is best suited to match the overall database [9]. The last step compares the data sub-tables calculated by each hash table with the overall database, selects the data within the threshold for matching, and finds the value with the smallest threshold as the output object. The biggest advantage of the MLH algorithm is that the application of the DCT algorithm can improve the operation efficiency of the algorithm, and the matching accuracy can be improved when multiple comparisons are performed at the same time to select the one with the smallest threshold. The schematic diagram of the MLH algorithm is shown in the following figure：

DOI: 10.54254/2755-2721/5/20230572

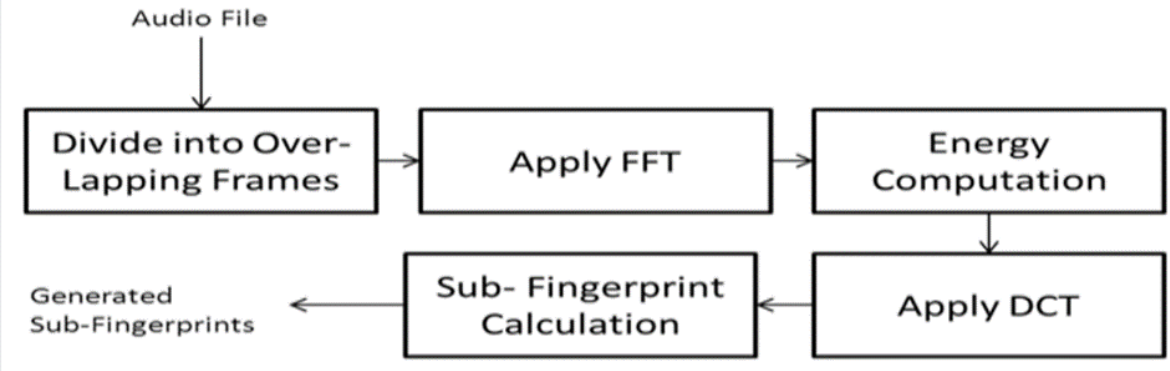

Figure 2. Multiple Hashing Algorithm how to work [10].

4. Comparison of Two Algorithms

4.1 Experimental design

The experiment will select four different audio types of namely normal sound, classical rock, and pop. For each audio type, four groups of audios with different complexity are selected for experimentation. Let the two algorithms identify them in order from hard to easy. The experiment judges the accuracy of the two algorithms by calculating the recognition rate. Since Philips Robust Hashing (PRH) algorithm does not generate sub-fingerprint bands, the sub-fingerprint bands are marked as 0. The Multiple Hashing Algorithm (MLH) is tested through 1, 2, 3, and 4 fingerprint bands. The experimental results are shown in the following four tables:

Table 1. The table of overall recognition rates.

Overall recognition rates					
Query Set	PRH	MLH			
	0	1	1,2	1,2,3	1,2,3,4
Set 1	98.43	98.71	99.13	99.36	99.95
Set 2	93.56	94.00	97.05	97.70	98.63
Set 3	97.03	97.47	98.47	98.80	99.63
Set 4	35.15	38.05	52.63	60.97	69.88

Table 2. The table of recognition rates for rock music.

Recognition rates for rock music					
Query Set	PRH	MLH			
	0	1	1,2	1,2,3	1,2,3,4
Set 1	99.67	100.00	100.00	100.00	100.00
Set 2	95.84	96.48	99.03	99.79	99.79
Set 3	99.21	98.77	99.54	99.79	100.00
Set 4	34.12	33.89	50.12	60.81	71.03

Table 3. The table of recognition rates for classical music.

Recognition rates for classical music					
Query Set	PRH	MLH			
	0	1	1,2	1,2,3	1,2,3,4
Set 1	96.21	97.23	98.19	98.66	99.60
Set 2	90.56	92.55	94.43	95.37	97.48
Set 3	94.02	95.16	97.06	97.48	99.60
Set 4	36.21	42.32	55.23	60.80	68.61

DOI: 10.54254/2755-2721/5/20230572

Table 4. The table of recognition rates for pop music.

Recognition rates for pop music					
Query Set	PRH	MLH			
	0	1	1,2	1,2,3	1,2,3,4
Set 1	97.13	99.21	99.78	100.00	100.00
Set 2	91.64	94.03	97.64	98.46	98.43
Set 3	98.46	98.73	98.69	99.25	99.25
Set 4	35.47	37.57	51.97	60.67	70.12

4.2 Experimental outcome

In four tables, the two algorithms are compared, and the performance of the Multiple Hashing Algorithm (MLH) when there is only one sub-fingerprint band is not much different from that of the Philips Robust Hashing (PRH) Algorithm. Both have high accuracy for the third and first sets of data. With the increase of sub-fingerprint bands, the accuracy of the MLH algorithm is gradually improved. In particular, the accuracy of the fourth group of the most difficult data is greatly improved. However, the increase of the sub-fingerprint band means that the calculation load and calculation time need to be increased, and the accuracy of MLH is proportional to the time. This means that the PRH algorithm should be used when processing simple audio data to speed up the computation time. When dealing with complex data, the MLH algorithm should be used to improve the accuracy and ensure that the user finds the desired song.

5. Conclusion

With the experimental results, the Multiple Hashing Algorithm (MLH) can improve the accuracy of complex audio information extraction and database comparison by increasing the number of sub-fingerprint bands but increasing the sub-fingerprint bands increases the running time of the algorithm and the information storage space. The Philips Robust Hashing (PRH) Algorithm has a high accuracy rate when processing simple audio information. The main reason for this result is the application of the Discrete Cosine Transform algorithm to the MLH algorithm, which helps the MLH algorithm group the received audio information in a more detailed way and split them into different sub-fingerprint bands during the calculation. bands. Each sub-fingerprint band is compared with the database to obtain a more accurate match. However, the more detailed information segmentation results in time wastage and requires more time to complete the information matching. Therefore, users should use the PRH algorithm when using an Audio fingerprint system for simple audio to reduce the system runtime. When processing complex audio, the MLH algorithm with multiple sub-fingerprint bands should be used to ensure the accuracy of the system operation. Users are more willing to spend more time getting the correct information than getting the wrong information. The findings of this experiment are very beneficial to companies and organizations that use the Audio fingerprint system in large numbers. These companies and organizations can use the optimized algorithm to improve their efficiency in processing audio information and thus provide better services to their customers. Although this experiment has obtained some data to determine the advantages and disadvantages of the two algorithms. However, this experiment did not collect specific time data on the operation of the algorithms. This makes it impossible to get more detailed conclusions in this paper. In the future, more experiments need to be conducted to distinguish the advantages and disadvantages of the two algorithms in more detail to further optimize the Audio fingerprint system.

References

[1] Doets, P. J. O. (2010). Modeling Audio Fingerprints: Structure, Distortion, Capacity.

[2] Haitsma, J., & Kalker, T. (2002, October). A highly robust audio fingerprinting system. In Ismir (Vol. 2002, pp. 107-115).

[3] Balado, F., Hurley, N. J., McCarthy, E. P., & Silvestre, G. C. (2007). Performance analysis of robust audio hashing. IEEE Transactions on Information Forensics and Security, 2(2), 254-

DOI: 10.54254/2755-2721/5/20230572

266.
[4] Schwarz, D. (2000, December). A system for data-driven concatenative sound synthesis. In Digital Audio Effects (DAFx) (pp. 97-102).
[5] Zhang, H., Liu, L., Long, Y., & Shao, L. (2017). Unsupervised deep hashing with pseudo labels for scalable image retrieval. IEEE Transactions on Image Processing, 27(4), 1626-1638.
[6] Heckbert, P. (1995). Fourier transforms and the fast Fourier transform (FFT) algorithm. Computer Graphics, 2, 15-463.
[7] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. IEEE Transactions on Computers, 100(1), 90-93.
[8] Lavner, Y., Cohen, R., Ruinskiy, D., & IJzerman, H. (2016, November). Baby cry detection in the domestic environment using deep learning. In the 2016 IEEE international conference on the science of electrical engineering (ICSEE) (pp. 1-5). IEEE.
[9] Balado, F., Hurley, N. J., McCarthy, E. P., & Silvestre, G. C. (2007). Performance analysis of robust audio hashing. IEEE Transactions on Information Forensics and Security, 2(2), 254-266.
[10] Kekre, H. B., Bhandari, N., Nair, N., Padmanabhan, P., & Bhandari, S. (2013). A review of audio fingerprinting and comparison of algorithms. International Journal of Computer Applications, 70(13).