Audio Search Algorithm

Submitted to: Engr Kaleem Ullah

Muhammad Bilal, Hassan Rizwan, Syeda Fatima Zahra
School of Electrical Engineering and Computer Science, NUST
CMS ID: 331538, 335753, 334379
Email: {mbilal, hrizwan, szahra}.bee20seecs@seecs.edu.pk

Abstract—**We have successfully developed a versatile audio search project that exhibits several desirable features. The algorithm is designed to be resilient against noise and distortion, ensuring accurate identification of short music segments captured via cellphone microphones. It can effectively handle the challenges posed by voice codec compression and operates efficiently on a vast scale, accommodating a database with over a million tracks. By utilizing a unique approach based on combinatorially hashed time-frequency constellation analysis of the audio, the algorithm offers remarkable characteristics, including transparency. This means that even when multiple tracks are mixed together, the engine can identify each individual track. Additionally, for applications such as radio monitoring, the search process is incredibly fast, deteting the songs added to the database; in our case, we added three songs to the database.**

I. INTRODUCTION

The algorithm had to be able to recognize a short audio sample of music that had been broadcast, mixed with heavy ambient noise, subject to reverb and other processing, captured by a little cellphone microphone, subjected to voice codec compression, and network dropouts, all before arriving at our servers. The algorithm also had to perform the recognition quickly over a large database of music with nearly 2M tracks, and furthermore have a low number of false positives while having a high recognition rate.

The Shazam algorithm can be used in many applications besides just music recognition over a mobile phone. Due to the ability to dig deep into noise we can identify music hidden behind a loud voiceover, such as in a radio advert. On the other hand, the algorithm is also very fast and can be used for copyright monitoring at a search speed of over 1000 times realtime, thus enabling a modest server to monitor significantly many media streams. The algorithm is also suitable for content-based cueing and indexing for library and archival uses.

The number of music files that could be detected have been restricted to a limited number. This avoids the arrival of any errors that come up with more files and also are enough files to practically demonstrate the working of this project.

II. BASIC PRINCIPLE OF OPERATION

Each audio file is “fingerprinted,” a process in which reproducible hash tokens are extracted. Both “database” and “sample” audio files are subjected to the same analysis. The fingerprints from the unknown sample are matched against a large set of fingerprints derived from the music database. The candidate matches are subsequently evaluated for correctness of match. Some guiding principles for the attributes to use as fingerprints are that they should be temporally localized, translation-invariant, robust, and sufficiently entropic. The temporal locality guideline suggests that each fingerprint hash is calculated using audio samples near a corresponding point in time, so that distant events do not affect the hash. The translation in variant aspect means that fingerprint hashes derived from corresponding matching content are reproducible independent of position within an audio file, as long as the temporal locality containing the data from which the hash is computed is contained within the file. This makes sense, as an unknown sample could come from any portion of the original audio track. Robustness means that hashes generated from the original clean database track should be reproducible from a degraded copy of the audio.

Furthermore, the fingerprint tokens should have sufficiently high entropy in order to minimize the probability of false token matches at non-corresponding locations between the unknown sample and tracks within the database. Insufficient entropy leads to excessive and spurious matches at non-corresponding locations, requiring more processing power to cull the results, and too much entropy usually leads to fragility and non-reproducibility of fingerprint tokens in the presence of noise and distortion.

III. COMPONENTS:

A. ROBUST CONSTELLATIONS

A time-frequency point is a candidate peak if it has a higher energy content than all its neighbors in a region centered around the point. Candidate peaks are chosen according to a density criterion in order to assure that the time-frequency strip for the audio file has reasonably uniform coverage. The peaks in each time-frequency locality are also chosen according amplitude, with the justification that the highest amplitude peaks are most likely to survive the distortions.

B. FAST COMBINATORIAL HASHTINGS

Fingerprint hashes are formed from the constellation map, in which pairs of time-frequency points are combinatorially associated. Anchor points are chosen, each anchor point having a target zone associated with it. Each anchor point is sequentially paired with points within its target zone, each pair yielding two frequency components plus the time difference between the points. These hashes are quite reproducible, even in the presence of noise and voice codec compression. Furthermore, each hash can be packed into a 32-bit unsigned integer. Each hash is also associated with the time offset from the beginning of the respective file to its anchor point, though the absolute time is not a part of the hash itself.

C. SEARCHING AND SCORING

To perform a search, the above fingerprinting step is performed on a captured sample sound file to generate a set of hash:time offset records. Each hash from the sample is used to search in the database for matching hashes. For each matching hash found in the database, the corresponding offset times from the beginning of the sample and database files are associated into time pairs. The time pairs are distributed into bins according to the track ID associated with the matching database hash.

Note that the matching and scanning phases do not make any special assumption about the format of the hashes. In fact, the hashes only need to have the properties of having sufficient entropy to avoid too many spurious matches to occur, as well as being reproducible. In the scanning phase the main thing that matters is for the matching hashes to be temporally aligned.

IV. RESULTS

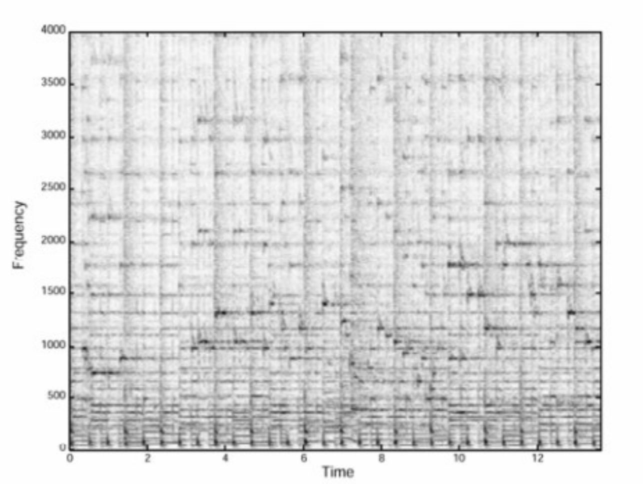

Figure 1 Spectrogram Obtained

The output from the algorithm was the spectrogram as can be seen above. The shading lines blur out the real detail in this spectrogram, which are the dots. The pattern of dots should be the same for matching segments of audio. If you put the constellation map of a database song on a strip chart, and the constellation map of a short matching audio sample of a few seconds length on a transparent piece of plastic, then slide the latter over the former, at some point a significant number of points will coincide when the proper time offset is located and the two maps are aligned in register.

So, the spectrogram of the audio that is inserted into the algorithm will be compared with the spectrograms that are already generated from the music files that are stored in the database. Over on over comparison, will indicate if the audio file does match to any of the pre-existing files and if it does, to which file. The number of matching points will be significant in the presence of spurious peaks injected due to noise, as peak positions are relatively independent; further, the number of matches can also be significant even if many of the correct points have been deleted.

The system can be improved by creating a constellation map of this spectrogram, and comparing it instead of the spectrogram. It will be more accurate. This can be a future improvement in the project.

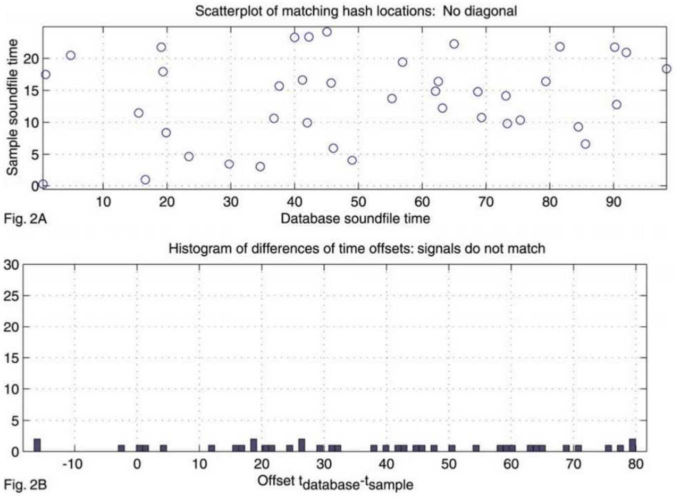

Figure 2 Matching hash locations scatterplot

This is a scatterplot of all the matching hash locations. It illustrates the relationship between database time and sample time for a track that does not match the sample. There are a few chance associations, but no clear linear correspondence appears to exist.

It's important to note that the matching and scanning phases in this process do not rely on any specific assumptions about the format of the hashes. In fact, the hashes only need to possess certain properties, such as having sufficient entropy to prevent an excessive number of false matches, and being reproducible. During the scanning phase, the key factor is temporal alignment of the matching hashes, ensuring they are synchronized in time.

V. LIMITATIONS

The algorithm was designed with a specific focus on recognizing sound files that already exist within the database. Its primary purpose is not to generalize to live recordings. Conversely, the algorithm demonstrates a remarkable sensitivity to the specific version of a track that has been sampled. Even when multiple performances of the same song by an artist are virtually indistinguishable to the human ear, the algorithm can accurately identify the correct one.

It's worth noting that there are instances where the algorithm is not technically incorrect, as it may identify examples of "sampling" or plagiarism. As mentioned earlier, there exists a tradeoff between true hits and false positives, leading to the selection of a maximum allowable percentage of false positives as a design parameter tailored to the application's requirements. This consideration ensures that the algorithm's performance aligns with the intended purpose.

VI. CONCLUSION

This project aimed to show a sized-down version of a full audio search algorithm. The developed project took a limited number of music files to make the project feasible and avoid unnecessary complications. For the quantity and other parameters that we chose, the project delivered outstanding results.

It was able to do both of its tasks correctly: audio fingerprinting and audio matching. Both were done with good accuracy and feasible results were obtained. Further work can be carried out in this line. It can, but is not limited to, adding more music files to make the database of to-be-detected music greater. The fingerprinting ability can be enhanced to better analyze the audio as it will help it in mapping over the files in the database.

VII. ACKNOWLEDGEMENTS

- Reinhard Sonnleitner, PhD Thesis
- Audio Identification via Fingerprinting Achieving Robustness to Severe Signal Modifications Sébastien Fenet, PhD Thesis
- Audio-Fingerprints And Associated Indexing Strategies For The Purpose Of Large-Scale Audio-Identification Avery Li-Chun Wang