ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

Ω.Π.Μ.Σ.
ΤΕΧΝΗΤΗ
ΝΟΗΜΟΣΥΝΗ

DEMOKRITOS
ΕΚΕΦΕ ΔΗΜΟΚΡΙΤΟΣ
ΙΝΣΤ. ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Landmark-based Audio Fingerprinting Using "audfprint"

Leandros Atherinis-Spartiotis
MTN 2301

Orestis Vaggelis
MTN 2306

Athens
June 2024

Table of Contents

1. Introduction

The proliferation of digital media has significantly increased the availability and accessibility of audio content. With platforms such as YouTube hosting vast amounts of audio and video material, identifying and managing copyrighted content has become increasingly challenging. Audio fingerprinting, a technique used to identify audio content through unique characteristics, offers a solution to this problem.

This project employs a landmark-based audio fingerprinting approach using the Audfprint library to determine the presence of specific songs within YouTube videos. Our system processes a given YouTube URL, extracting audio features and comparing them to a pre-existing database of fingerprints. This method enables efficient and accurate identification of audio segments, even in the presence of noise or distortion.

In this paper, we detail the design, implementation, and evaluation of our audio fingerprinting system. We begin by discussing related work and the theoretical foundations of audio fingerprinting, followed by a description of our methodology. We then describe the end-to-end system created for the task and present the results of our system's performance, demonstrating its capability in identifying known audio content within a variety of YouTube videos. Finally, we conclude with a discussion of the implications of our findings and potential future enhancements.

2. Audio Fingerprinting

Audio fingerprinting is a technique used to identify audio content by analyzing unique characteristics of a sound signal. Unlike metadata tagging, audio fingerprinting generates a distinctive representation of the audio that allows for efficient and accurate identification.

2.1. Common Approaches

The most common approach to audio fingerprinting involves transforming the audio signal into a spectral representation, typically through a short-time Fourier transform (STFT). This representation captures frequency and time information, creating a spectrogram. Key features are extracted from this spectrogram, such as peaks in the frequency domain, which are robust to distortions like noise and compression.

These features are encoded into compact fingerprints, which are stored in a database. When an audio query is provided, its fingerprints are generated and compared against the database to identify matches. This method is effective for music recognition, copyright detection, and content-based audio retrieval.

3. Landmark Based Audio Fingerprinting

Landmark-based audio fingerprinting is a robust method for identifying audio content by focusing on distinctive features within the audio signal. This approach enhances traditional methods by identifying specific points, or landmarks, within the audio's spectrogram that are highly resilient to noise and distortions.

3.1. Methodology

In this approach, the audio signal is first transformed into a spectrogram, providing a time-frequency representation. The system then detects prominent peaks within this spectrogram, which serve as landmarks. These peaks are selected based on their amplitude and relative position, ensuring they remain identifiable even when the audio undergoes various transformations.

3.2. Feature Extraction

The landmarks are paired based on their temporal and frequency distances, creating unique pairs that form the basis of the audio fingerprint. Each pair represents a robust feature that can be matched against a database of existing fingerprints. This method allows for efficient comparison, as only the landmarks need to be analyzed.

3.3. Advantages

This technique provides several advantages, including robustness to common audio alterations like compression, equalization, and environmental noise. It also allows for efficient storage and retrieval, as the fingerprints are compact and easily searchable.

3.4. Applications

Landmark-based audio fingerprinting is particularly effective in environments where audio quality varies, such as user-generated content on platforms like YouTube. Its precision and efficiency make it suitable for applications in music recognition, copyright management, and audio content analysis.

4. Audfprint

Landmark-based audio fingerprinting in audfprint involves identifying unique and robust features in an audio signal to create a distinctive fingerprint that can be used to recognize and match audio clips despite noise and distortions. The process starts by converting the audio to a standardized format, such as mono, and resampling it to a consistent sampling rate.

The audio signal is then transformed into a spectrogram using the Short-Time Fourier Transform (STFT). The STFT involves breaking the audio signal into overlapping short segments, or frames, and computing the Fourier Transform for each frame. This transform converts the time-domain signal into the frequency domain, representing how the frequency content of the signal changes over time.

A spectrogram is a visual representation of the spectrum of frequencies in a signal as it varies with time. It displays time on the horizontal axis, frequency on the vertical axis, and the amplitude (or energy) of each frequency component as color intensity. The result is a 2D image where peaks in the spectrogram indicate points of high energy at specific times and frequencies. In the spectrogram, significant peaks, or landmarks, are identified where the energy is concentrated. These peaks are selected based on their local prominence compared to the surrounding areas in the spectrogram.

These identified peaks are paired to form time-frequency pairs, and each pair is encoded into a compact 20-bit hash. The encoding captures the frequency of the first peak, the frequency difference to the second peak, and the time difference between the peaks. This hash effectively captures the unique relationship between pairs of peaks, creating a distinctive fingerprint for each segment of the audio. These fingerprints are stored in a hash table, where each hash is associated with the corresponding audio file and its time offset. The hash table uses 32-bit entries that combine the absolute time and track ID, enabling efficient storage and retrieval.

During querying, the same fingerprinting process is applied to the query audio. The resulting hashes are matched against the database to find the best matches based on the number of common hashes and their alignment. The matching process involves comparing the hashes from the query with those in the database, aligning them based on their time offsets, and counting the number of matches. Matches are ranked by the number of aligned hashes, ensuring robust and accurate identification of audio clips, even in challenging conditions.

5. Implementation

To test our implementation of the system described above we create a dataset of 230 songs in .mp3 form and we extracted and stored their fingerprints in a pklz file which seemed contribute to the inference speeds.

We also created an app with two ways to input sound for inference. For the first app you input a youtube url, the system isolates and downloads the sound of that video in .mp3 form, then it extracts its features and compares it with the fingerprints on our database. It then returns the closest result if the difference is within the threshold.

For the second app the system records a sound sample of 5 second duration, mono channel and 44100Hz sampling rate. Then proceeds to do the same as the other implementation and returns the closest result.

6. Results and Evaluation

For our test dataset we found some songs that are on the background of movie scenes to test how peak changes like speech and gunshots affect the system's ability to identify the songs.

For example, the database contains the original recording of Queen's We Will Rock You, so we found the intro scene of A Knight's Tale with Heath Ledger that has an altered version of the song but with the same vocals at most points also has background noise and speech and inference was accurate.

Link:

In another test case we found the opera scene from Mission Impossible 6 that has Nessun Dorma by Giacomo Puccini on the background with great variance on the volume as its serves as background sound and also as main at many points. The scene consists of many peaks that aren't originated from the song like sound effects speech and gunshots but also the song is a different version sang by a different tenor and could be in a slightly different key so we believe that this also contributes to the inability of the system to accurately find which song it is even though the Luciano Pavarotti version is included in the database with great sound quality.

Link:

We also tried both test with the record sound implementation and included a very characteristic part of each song that in both videos was a main sound and still the same results.

We also observed higher accuracy when tested in parts with vocals (with the 5 second recording) and also in songs with a loud and clear beat.

We tested Dejavu and also tried comparing the linear distance of mid-term features extracted by pyAudioanalysis and concluded that audfprint had higher accuracy and lower inference times in most cases.

Bibliography

[1] https://www.ee.columbia.edu/~dpwe/LabROSA/matlab/fingerprint/

[2] https://github.com/dpwe/audfprint

[3] https://www.ee.columbia.edu/~dpwe/resources/matlab/audfprint/

[4] https://github.com/shazamio/ShazamIO

[5] https://github.com/worldveil/dejavu