MUSIC FINGERPRINTING

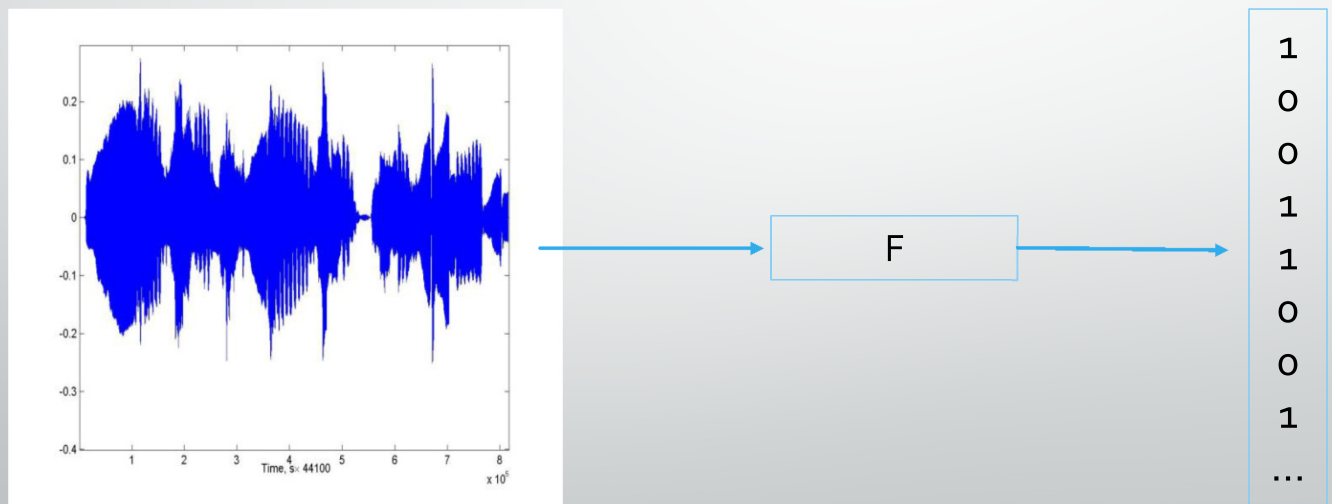
in MUSIC INFORMATION RETRIEVAL

Music Information Retrieval

A field of science for retrieving the rhythmic features from music. A combination of *Signal Processing*, *Machine Learning*, *Music Study*, *Audio Mining*, *Music Notation*, *Artificial Intelligence*.

AUDIO FINGERPRINTING

An ***Audio fingerprint*** is a condensed digital summary, deterministically generated from an audio signal that can be used to identify an audio sample or quickly locate similar items in an audio database.



Audio Fingerprint v/s Cryptographic hash functions

- **Mathematical Equivalence v/s Perceptual similarity**
  Assume X and Y are two objects that are mapped into H(X) and H(Y) by a crypto. hash function H. Strict mathematical equality of H(X) and H(Y) implies an equality of X and Y with a very low probability of error. In case of audio, we are not interested in strict mathematical equivalence but perceptual similarity.

- **Transitivity property**
  If two sound tracks X and Y are perceptually similar while Y and Z are perceptually similar to each other, it does NOT imply that X and Z are perceptually similar. Transitivity property essentially holds for all mathematical hash functions.

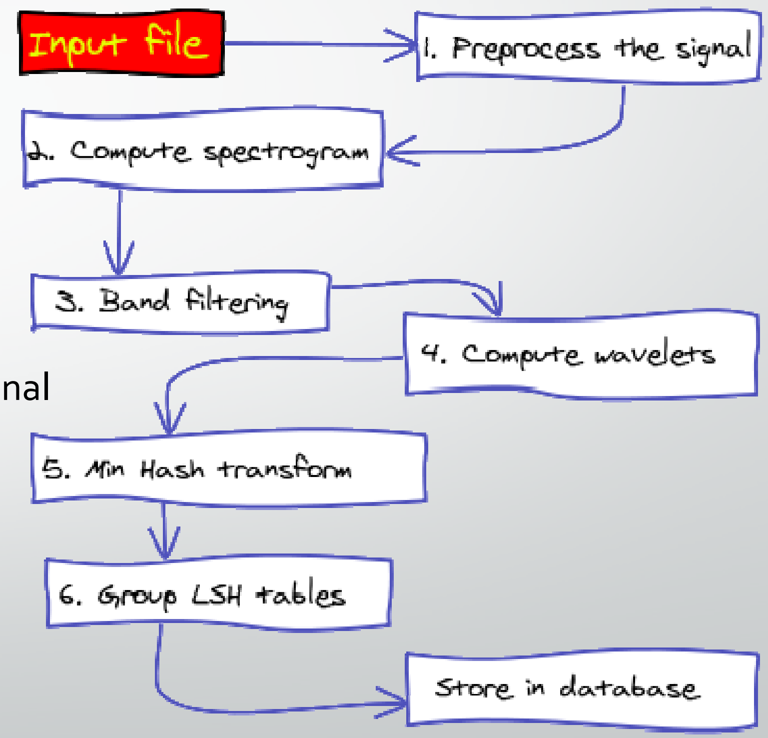
Therefore, in stead of mathematical equivalence, we use threshold comparisons:

$|F(x) - F(y)| \leq T$: implies X and Y are similar

$|F(x) - F(y)| > T$: implies X and Y are not similar

Outline

- Idea
  - An attempt to mimic human music recognition abilities
- Audio Fingerprint
  - Unique identifier of an audio signal
  - Content-based signature that summarizes an audio recording
  - Uses relevant (perceptual) acoustics characteristics of signal
- Fingerprinting System
  - Database of known fingerprints
  - Query system



Points To Be Considered

- Accuracy
  - Function of correct, missed, and wrong identifications
- Complexity
  - Computational costs of fingerprint extraction, size of fingerprint, search complexity, comparison complexity, etc.
- Robustness
  - Ability to accurately identify an item (no matter how compressed or distorted it is)
- Granularity
  - Ability to identify a signal from a short excerpt
- Scalability
  - Performance with very large databases

Audio Identification Framework

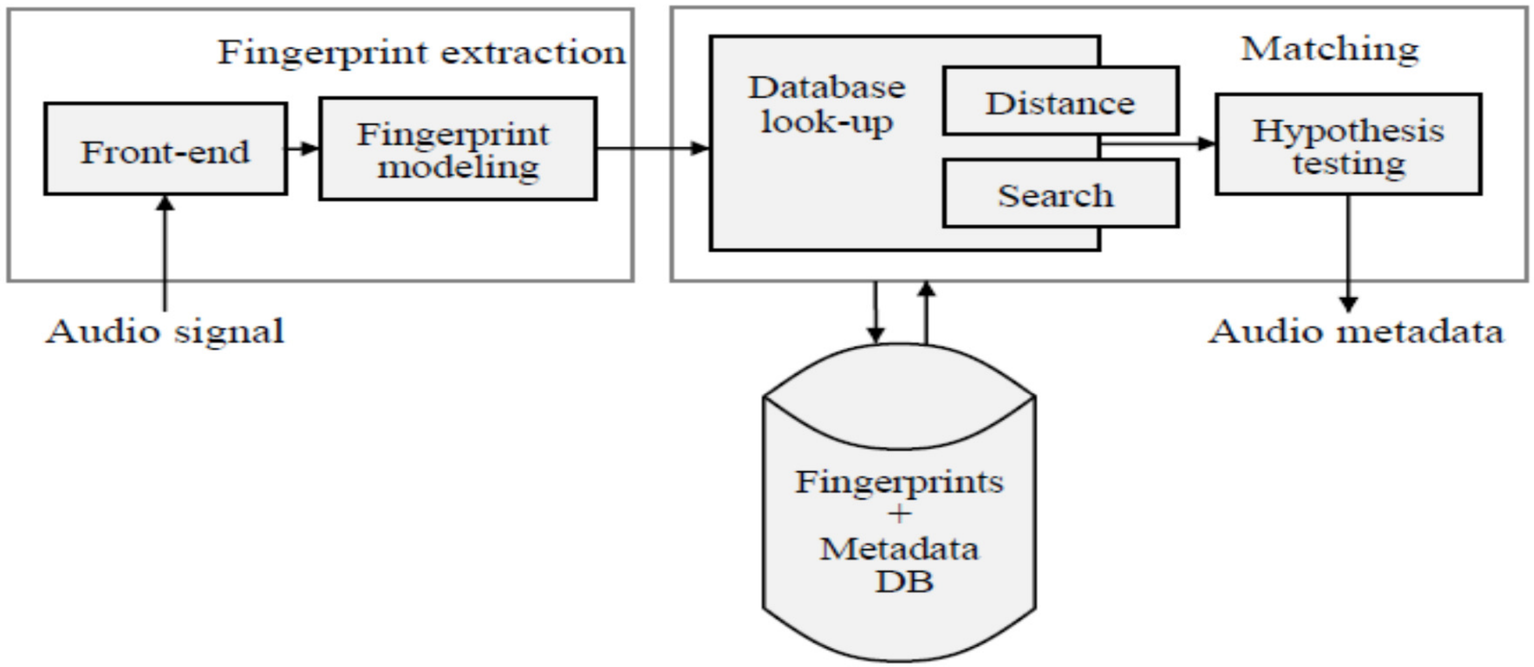
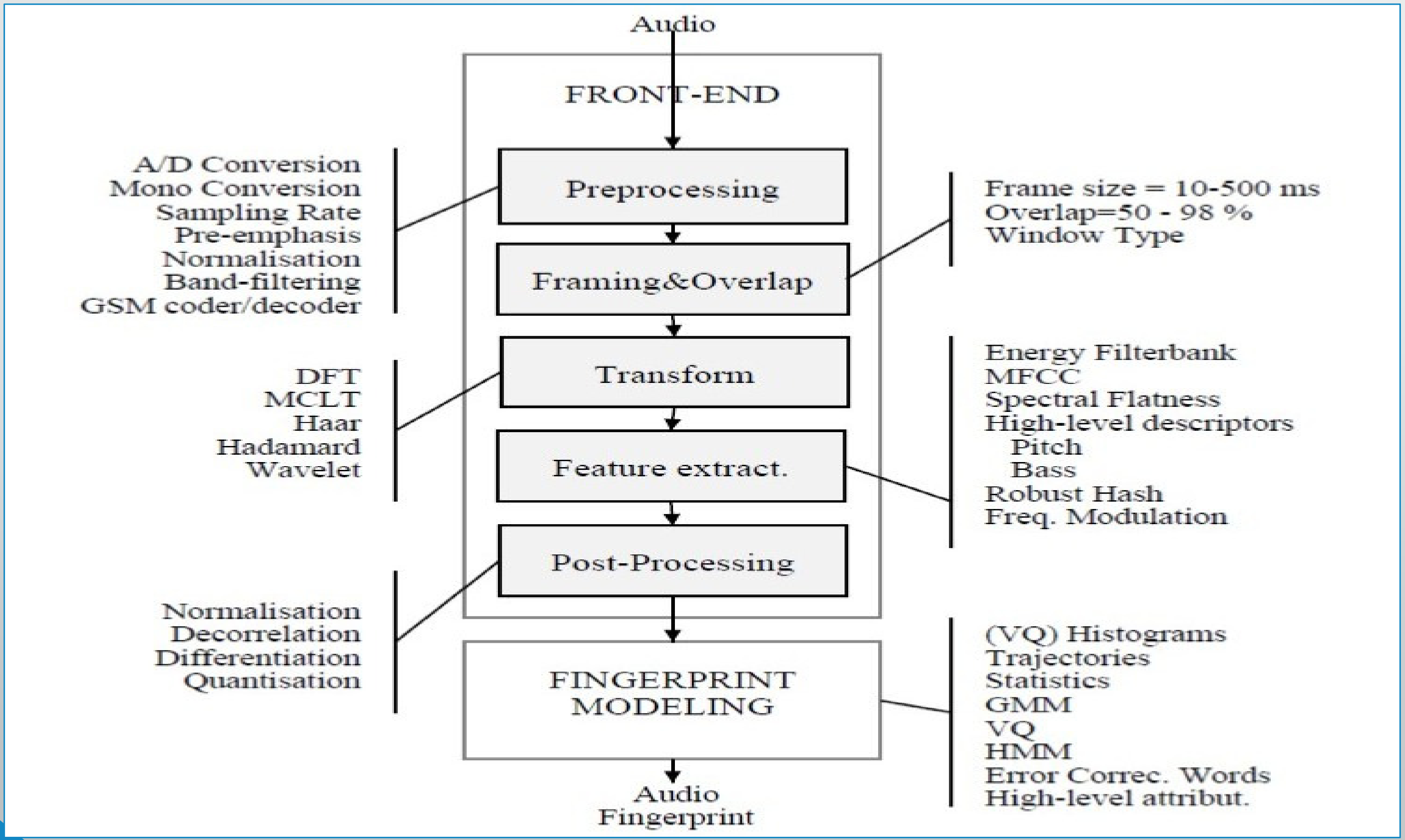


Fig. 1. Content-based Audio Identification Framework.

Fingerprint Extraction Framework



Methodology

Audio Identification Framework

- Input: An Audio Sample
- Output: Audio MetaData (Matched Fingerprint)
- Processing:

  - > Preprocessing
  - > Fingerprint Extraction
  - > Database Lookup
  - > Distance Search
  - > Audio Metadata

Fingerprint Extraction Framework

- Input: An Audio File
- Output: Fingerprint Hash (Digest)
- Processing:

  - > Segmentation
  - > Compute Spectrogram
  - > Feature Extraction
  - > Locality Hashing
  - > Fingerprint Modelling

Block by Block Extraction Process

- For Fingerprints, the Input has to be a complete song file so that all the ***distinct patterns and rhythmic information*** can be retrieved.
- The input file will be *segmented* into *smaller and distinct segmental patterns* and a list of those patterns will be returned.
- The segmented pattern list will be used to *compute and plot the* **Spectrograms**.
- Also, the above list will be used to *extract the rhythmic features* from the *segments* of the song.
- A *Hashing Mechanism* known as **Locality Sensitive Hashing** will be applied on the features to *get the digest* of the patterns.
- At Last, *Fingerprint Modelling* will be done to *store the unique digest* (hash) in our database.
- The following *features* will be retrieved for *Fingerprinting*:

Zero Crossing Rate, Sample Rate, Spectral Centroid, Chromagram, Onset Rate, LogAmplitude

Block by Block Identification Process

- The Input will be either a direct audio file or through speech recognition on the real time (it can be a complete song or minimally must be a 10 sec long clip).
- The *Fingerprint Extraction Framework* will be used to extract the rhythmic features of the audio file provided.
- The *Database Lookup and Match* will be performed for *audio fingerprint* extracted above with the existing fingerprints in the database.
- The *Matching* will involve a Nearest Neighbor Search to identify the *closest and most similar* fingerprint existing in the database.
- The above step will return a *list of song fingerprint* matched with their corresponding accuracy in the ascending order.
- The fingerprint with the *higher accuracy* will be considered as the matched item, and hence will be returned with its *Metadata.*

Existing Applications

- Example Systems

  - Shazam

  - SoundHound

- Locality Sensitive Hashing (LSH)

  - Winner Take All Hash

  - MinHash

Lets Talk Shazam

- Recognize a song from a short snippet of audio recorded on a mobile phone.
  - –Database of nearly 2 million tracks
  - –Recorded snippet up to 15 seconds in length
- “Combinatorically hashed time-frequency constellation analysis”

Applications

- Identify Songs, Melodies
- Detect Cover Songs
- Group Similar Songs
- Build Song Playlist

What makes us different?

All the existing applications like *Shazam* and *Soundhound* is about exact audio song match, but our program not only matches the same song but also its *similar songs* like cover and duplicates, and also sort them according to their ***similarity percentage***, which serves as a variety to the result.

References

- Jaap Haitsma and Ton Kalker, "A highly robust audio fingerprinting system", Philips Research , Eindhoven, The Netherlands, October 2001
- Music IP corporation, Available HTTP: *musicip.com*
- Haitsma J., Kalker T. and Oostveen J., "Robust Audio Hashing for Content Identification", Content Based Multimedia Indexing 2001, Brescia, Italy, September 2001.