CHAPTER 9

AUDIO FINGERPRINTING

Fingerprinting aims at identifying audio recordings in a previously generated database. More specifically, each recording is represented by a fingerprint, a unique and compact digest summarizing the (perceptually) relevant aspects of the recording. The fingerprint is also referred to as *perceptual hash*. A database containing previously extracted fingerprints can be used to identify an unknown recording. In contrast to most of the other systems presented in the book, fingerprinting does not attempt to extract musical properties from the audio signal but aims at identifying a specific recording as opposed to a specific song. Different music performances (or recordings) of the same song should therefore have different fingerprints. However, a recording still has to be identified when subjected to quality degradation such as perceptual audio coding, added noise, distortions, and other typical signal manipulations.

There are two main areas of application: *broadcast monitoring* allows rights holders the verification of paid royalties and end consumer apps enable the user to either easily identify music or to make use of added value services offering extra information for a song such as the album cover image or tags and other meta data of interest. Cano et al. give a detailed overview of various other applications for which fingerprinting can be of use [367].

Fingerprinting is not to be confused with *watermarking*; the latter embeds a perceptually unnoticeable data block directly in the audio data, utilizing methods similar to perceptual audio coding. Watermarking thus enables the content provider to embed different watermarks in the same audio content. It allows, for example, to embed a user-specific watermark in the specific copy of the recording in order to identify this specific user copy of the recording later. This is not possible with fingerprinting. Watermarking also allows to

An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics, First Edition. Alexander Lerch.

Table 9.1 Main properties of fingerprinting and watermarking in comparison

Property	*Fingerprinting*	*Watermarking*
Allows Legacy Content Indexing	+	–
Allows Embedded (Meta) Data	–	+
Leaves Signal Unchanged	+	–
Identification of	Recording	User or Interaction

embed meta data directly — the user has direct access to this additional data (e.g., song title or artist name) while with fingerprinting he would have to rely on a database connection or local tags. The major disadvantage of watermarking is that the audio signal has to be modified. This can on the one hand possibly degrade the audio quality (with similar quality degradation as caused by perceptual encoders) and on the other hand cannot cover legacy audio recordings which have been either already distributed or through other (distribution) channels. Table 9.1 summarizes the main properties of fingerprinting and watermarking.

A fingerprinting system consists of two basic building blocks, the fingerprint extraction of the seed tracks and a database of previously extracted fingerprintings coupled with unique identifiers or additional meta data about the piece of music. Figure 9.1 visualizes these blocks; the upper part of the graph shows the process of adding new entries to the database (done by the service provider) and the lower part shows the query for a recording by a client.

The requirements on a general fingerprinting system have been summarized by Cano et al. [368] as:

- *Accuracy & reliability*: high number of correct identifications (TPs) compared to the number of missed identifications (FNs) and wrong identifications (FPs).
- *Robustness & security*: high accuracy even in case of a heavily distorted signal. Possible distortions include lossy compression, added noise, equalization, interference, and non-linearities of the transmission path. Sophisticated systems should also be robust against changes in tempo and pitch.
- *Granularity*: the shorter the length of an excerpt required for its identification the better (modern systems require a length of a few seconds).
- *Versatility*: independence of detection from the file format and the file origin as well as an application-independent implementation.
- *Scalability*: good performance on very large databases and a large number of simultaneous identification queries.
- *Complexity*: low computational cost of both extracting a fingerprint and finding this fingerprint in the database.

9.1 Fingerprint Extraction

Since the fingerprint should be robust against bandwidth restrictions and audio format, the two most common pre-processing steps are down-mixing to a single mono channel and

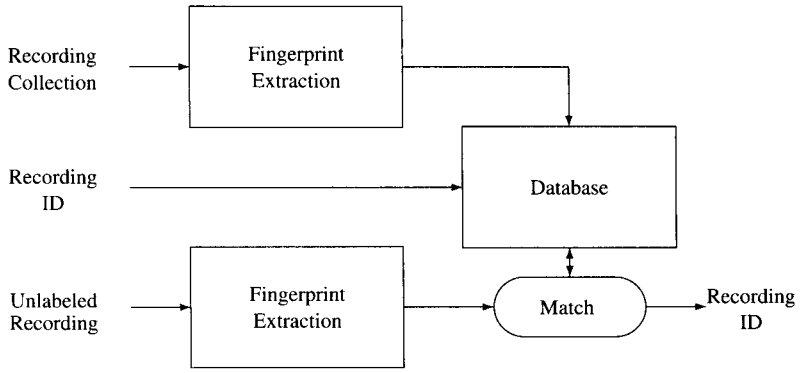

Figure 9.1 General framework for audio fingerprinting. The upper part visualizes the training phase and the lower part the query phase

down-sampling to a lower sample rate (usually 5–20 kHz). Applying a high-pass filter discards frequency components below the lowest transmittable frequency of phones in an optional pre-processing step.

The features of interest for the task of fingerprinting are robust to distortions, efficient to compute, and allow the unambiguous identification of the recording. While musical properties might help in the identification process, they are no necessity — low-level features generally should suffice.

A system that can be seen as an early predecessor of today's fingerprinting systems targeted the detection of advertisements in broadcast streams [369]. Lourens used the advertisement's energy envelope and selected a "unique" section serving as fingerprint. In most contemporary systems, the features are extracted in the frequency domain. These features include MFCCs [370, 371], a spectral flatness measure and a spectral crest factor per frequency band [372], a spectral centroid per subband [373], band energies [374] or (the sign of) energy band differences [375], carefully selected spectral peaks [376, 377], statistical moments of subbands [378], and modulation frequency features [379].

Frequently, the extraction process yields multiple features per block (usually each with a word length of 32 bits). In order to receive a compact information and to decrease the memory footprint many of these features (or the feature derivatives) are quantized into a binary or ternary representation.

The resulting fingerprint then contains a unique series of quantized feature values or feature vectors.

9.2 Fingerprint Matching

The extracted fingerprint, representing the unknown recording, has to be compared against all previously stored fingerprints in the database. The similarity (or distance) measure has to be fast for large databases. Common metrics include a correlation measure [380, 381], the Euclidean distance [373, 374, 382], and the Manhattan distance (which in the case of binary input equals the *Hamming distance*) [383, 384], but there exist many possible alternatives (see, e.g., [385]).

Even with a fast-to-compute similarity measure, it is not possible to compare every query fingerprint against all stored fingerprints in a large database due to workload and response time constraints. It is, for example, possible to pre-compute distances between the stored fingerprints in order to find different entry points for the query [374]. It is also possible to use different similarity measures, an efficient one to discard many database entries in a first run and a second more accurate similarity measure to be computed on the selected small subset [386].

There are many other ways to improve database performance; one example would be to pre-sort the database entries with their "popularity" in order to reduce search time for songs with frequent queries.

9.3 Fingerprinting System: Example

To allow a better understanding of the process of audio fingerprinting, a widespread and frequently referenced system will be explained in detail in the following. It is the Philips fingerprinting system as published by Haitsma et al. [384].

After the signal is down-mixed to one channel and down-sampled to a sample rate of 5 kHz, it is subjected to a (von-Hann-windowed) STFT. The block length is 0.37 s and the hop size is 11.6 ms. The large block overlap ratio increases the system's robustness against time-shift operations.

The magnitude spectrum is divided into 33 non-overlapping bands in the range 300–2000 Hz. The bandwidth is logarithmically increasing with frequency to take into account the non-linear frequency resolution of the human ear (see Sect. 5.1). The energy E per band with band index k is then used to derive a binary result by using both the time and frequency derivative:

$$v_{\mathrm{FP}}(k,n) = \begin{cases} 1 & \text{if } \big(\Delta E(k,n) - \Delta E(k,n-1)\big) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.1}$$

with

$$\Delta E(k,n) \quad = \quad E(k,n) - E(k+1,n). \tag{9.2}$$

This results in a 32-bit word per STFT; Haitsma et al. refer to this word as *subfingerprint*. One complete fingerprint consists of 256 subsequent subfingerprints and has thus a length of 3 s. Figure 9.2 shows an overview of the subfingerprint extraction.

The distance measure for the database search is the Manhattan distance; since the fingerprints are binary the Manhattan distance equals the *Hamming distance*. The length of 3 s appears to be sufficient for the identification of a song from the database. The database has to contain the series of all subfingerprints of each complete recording. Thus, if the database contains one million songs of approximately 5 min length, it holds more than 25 billion subfingerprints. Even in the case of a highly compressed subfingerprint format and the use of the computationally efficient Hamming distance, this amount of subfingerprints rules out the brute force approach of searching the whole database for each query.

Haitsma et al. suggested two methods to improve computational efficiency of the database search, a simple and a more refined method. First, a lookup table is added to the database. This table contains all possible 32-bit subfingerprints which leads to a maximum number of 2^{32} table entries. Each table entry points to a list of occurrences in the database. The lookup table can also be replaced by a hash table for efficiency.

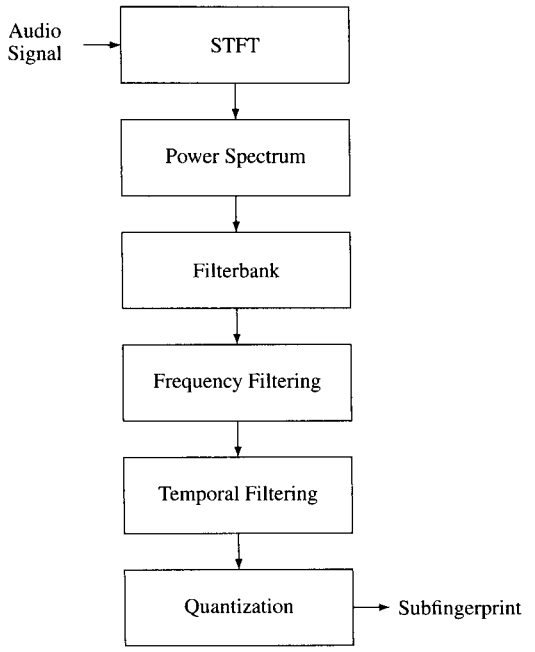

Figure 9.2 Flowchart of the extraction process of subfingerprints in the Philips system

The simple approach is based on the assumption that at least one of the 256 extracted subfingerprints has an exact match at the correct position in the database. Therefore, only the database entries listed under one of the 256 subfingerprints of the current query have to be evaluated as possible matches.

While this approach reduces the workload dramatically, the assumption that there is at least one subfingerprint without a bit error is only valid for audio with minor degradations. For highly distorted signals a larger number of bit errors can be expected. It is logical to assume bit errors in the subfingerprints. This has the disadvantage of drastically increasing the workload: if *one* bit error is expected per subfingerprint, the number of database queries and this the computational workload increases by a factor of 33. In order to reduce this additional workload while still taking into account possible bit errors, the concept of the *reliability* of a bit error is introduced in the enhanced system proposal. Since the bits of a subfingerprint are computed by energy differences, the likelihood of a bit being flipped (a bit error) is high for small energy differences and low for large differences. Thus, the bits can be ranked by their reliability and only the unreliable bits have to be flipped for the database search.