Automatic Recognition of Samples in Musical Audio

Jan Van Balen

MASTER THESIS UPF / 2011

Master in Sound and Music Computing.

Supervisors:
PhD Joan Serrà, MSc. Martin Haro
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

UNIVERSITAT
POMPEU FABRA

Acknowledgement

I wish to thank my supervisors Joan Serrà and Martin Haro for their priceless guidance, time and expertise. I would also like to thank Perfecto Herrera for his very helpful feedback, my family and classmates for their support and insightful remarks, and the many friends who were there to provide me with an excessive collection of sampled music. Finally I would like to thank Xavier Serra and the Music Technology Group for making all this possible by accepting me to the master.

Abstract

Sampling can be described as the reuse of a fragment of another artist's recording in a new musical work. This project aims at developing an algorithm that, given a database of candidate recordings, can detect samples of these in a given query. The problem of sample identification as a music information retrieval task has not been addressed before, it is therefore first defined and situated in the broader context of sampling as a musical phenomenon. The most relevant research to date is brought together and critically reviewed in terms of the requirements that a sample recognition system must meet. The assembly of a ground truth database for evaluation was also part of the work and restricted to hip hop songs, the first and most famous genre to be built on samples. Techniques from audio fingerprinting, remix recognition and cover detection, amongst other research, were used to build a number of systems investigating different strategies for sample recognition. The systems were evaluated using the ground truth database and their performance is discussed in terms of the retrieved items to identify the main challenges for future work. The results are promising, given the novelty of the task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sampling, as a creative tool in composition and music production, can be described as the reuse of a fragment of another artist's recording in a new work. The practice of digital sampling has been ongoing for well over two decades, and has become widespread amongst mainstream artists and genres, including pop and rock [7, 8]. Indeed, at the time of writing, the top two best selling albums as listed by the Billboard Album top 200 contain 8 and 21 credited samples, respectively[1] [9, 10, 11], and the third has already been sampled twice. However, in the Music Information Retrieval community, the topic of automatic sample recognition seems to be largely unaddressed [12, 13].

This project aims at developing an algorithm that can detect when one song in a music collection samples a part of another. An application of this that may be first thought of is the detection of copyright infringements. However, there are several other motivations behind this goal. A number of these are explained in section 1.1.

Even though cases of sampling can be found in several musical genres, this thesis will restrict to the genre of hip hop, to narrow down the problem and because hip hop as a musical genre would not exist as such without the notion of sampling. A historical and musicological context of sampling is given in section 1.2. Section 1.3 outlines the research and how it is reported on in the remainder of this document.

1.1 Motivation

A first motivation originates in the belief that the musicological study of popular music would be incomplete without the study of samples and their origins. Sample recognition provides a direct insight into the inspirations and musical resources of an artist, and reveals some details about his or her composition methods and choices made in the production.

[1] Game - *The R.E.D. Album* and Jay-Z & Kanye West - *Watch The Throne* (`www.billboard.com/charts/billboard-200`).

Figure 1.1 shows a diagram of sample relations between some of the artists appearing in the music collection that will be used for the evaluation part of this thesis (Chapter 3). The selection contains mostly artists that are generally well represented in the collection. The darker elements are sampled artists, the lighter elements are the artists that sampled them. The diagram shows how the sample links between artists quickly give rise to a complex network of influence relations.

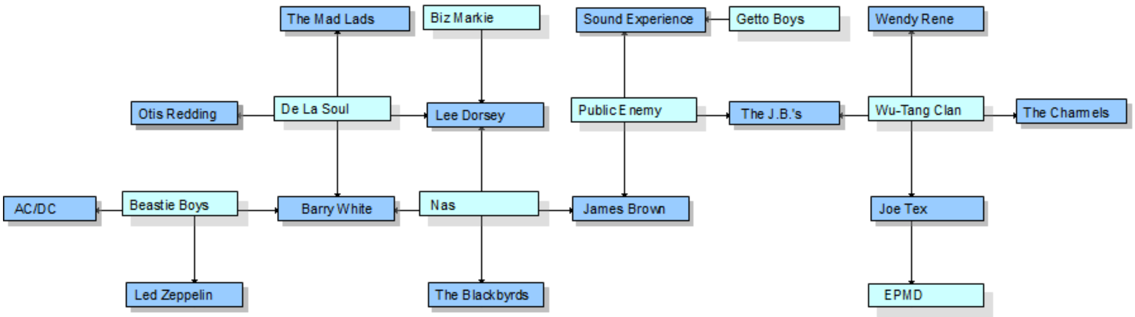


Figure 1.1: Network representation of a part of the music collection established for the evaluation methodology of this thesis. The darker elements are sampled artists, the lighter elements are the artists that sampled them.

However, samples also hold valuable information on the level of musical genres and communities, revealing influences and dependence. An example of this are researchers who have studied the way hip hop has often sampled 60's and 70's African-American artists, paying homage to the strong roots of black American music [7] and has often referred to icons of the African-American identity consciousness of the 1970's, for example by sampling soundtracks of so-called blaxploitation films, a genre of low-budget, black-oriented crime and suspense cinema [14].

Sample recognition can also be applied to trace musical ideas in history. Just like melodic similarity is used in the study of folk songs [15] and cover detection research [16], sample recognition could allow musical re-use to be observed further into the recorded musical history of the last two decades.

As an example of the complex history a musical idea can have, consider the popular 2006 Black Eyed Peas single *Pump It*. It samples the song *Misirlou* by Dick Dale (1962), pioneer of the surf music genre, though in the album credits, the writing is attributed to Nicholas Roubanis, a Greek-American jazz musician who made an instrumental jazz version of the song in 1941 [17]. The song is in fact a popular Greek folk tune, played for the first time by the Michalis Patrinos rebetiko band in Athens in 1927. Then again, the tune has more recently gained a completely different cultural connotation after the surf version of *Misirlou* was used in the opening scene of the popular 1994 Film *Pulp Fiction* by Quintin Tarantino. The above illustrates how one melody can have many different connotations and origins.

A third motivation is that sample recognition from raw audio provides a way to bring structure in large music databases. It could complement a great amount of existing research in the automatic classification of digital information. Like many systems developed and studied in information retrieval, music similarity and music recommendation, automatic classifiers are a more and more indispensable tool as the amount of accessible multimedia and the size of personal collections continue to grow [12, 18, 13]. Examples of such applications developed specifically in the field of content based Music Information Retrieval include automatic genre classification, performer identification and mood detection, too name a few. A good overview of directions and challenges in content-based music information retrieval is given by Casey et al. in [12] and Müller et al. in [13].

A third possible motivation is the use of automatic sample detection for legal purposes. Copyright considerations have always been an important motivation to understand sampling as a cultural phenomenon; a large part of the academic research on sampling is not surprisingly focused on copyright and law. In cases of copyright infringement, three questions classically need to be answered:

1. Does the plaintiff own a valid copyright in the material allegedly copied?
2. Did the defendant copy the infringed work?
3. Is the copied work substantially similar?

where the most difficult question is the last one [7]: the similarity of copied work is not only a matter of length and low-level musical context, but also of originality of the infringed work, and how important a role the material plays in both the infringing and the infringed work. Meanwhile, it is clear that even an ideal algorithm for sample detection would only be able to answer the second question. The use of the proposed sample detection algorithm for legal purposes is therefore still limited.

1.2 Musicological Context

1.2.1 Historical Overview

The Oxford Music Dictionary defines sampling as "the process in which a sound is taken directly from a recorded medium and transposed onto a new recording" [19]. As a tool for composition, it originated when artists started experimenting with tapes of previously released music recordings and radio broadcasts to make musical collages, as was common in musique concrète [14]. Famous early examples include the intro of The Beatles' *All You Need is Love* (1967), which features a recorded snippet of the French national hymn *Les enfants de la patrie.*

The phenomenon spread out when DJ's in New York started using their vinyl players to do what was already then being done by 'selectors' in Kingston, Jamaica: repeating and mixing parts of popular recordings to provide a continuous stream of music for the dancing crowd. Jamaican-born DJ Kool Herc is credited for being the first to isolate the most exciting instrumental break in a record and loop that section to obtain the 'breakbeat' that would later become the corner stone of hip hop music [20]. The first famous sample-based single was Sugarhill Gang's *Rapper's Delight* (1979), containing a looped sample taken from *Good Times* by Chic (1979) [19].

The big breakthrough of sampling, however, followed the invention of the digital sampler around 1980. Its popularisation as an instrument came soon after the birth of rap music, when producers started using it to isolate, manipulate and combine well-known and obscure portions of others recordings in ways it could no more be done by 'turntablists' using record players [21]. Famous examples of hip hop albums containing a great amount samples are *Paul's Boutique* by Beastie Boys, and *3 Feet High and Rising* by De La Soul (both 1989). The sampler became an instrument to produce entirely new and radically different sonic creations.

The possibilities that the sampler brought to the studio have played a role in the appearance of several new genres in electronic music, including house music in the late 90's (from which a large part of 20th century Western dance music originates), jungle (a precursor of drum&bass music), dub and trip hop [22]. A famous example of sampling in rock music is the song *Bittersweet Symphony* by The Verve (1997), which looped a pattern sampled from a 1966 string arrangement of The Rolling Stones' *The Last Time* (1965) [19].

1.2.2 Sampling Technology

Sampling can be performed in various ways. Several media have been used for recording, manipulation and playback of samples, and each medium has its on functionalities. The most important pieces of equipment that have been used for the production of a sample-based compositions are:

Tape players: The earliest experiments in the recycling of musical recordings were done using tape [23]. Recordings on tape could be physically manipulated between recording and playback. This freedom in editing and recombination has been explored in so-called tape music from the 1940's on. An examples of a notable composer working with tape was John Cage, whose *William's Mix* (1952)was spliced and put together from hundreds of different tape recordings [24].

Turntables: The birth of repetitive sampling, playing one sample over and over again, is attributed to Jamaican 'selectors' who, with their mobile 'sound systems', looped the popular sections of recordings at neighbourhood parties to please the dancing crowds. Several record labels even re-oriented to compete in producing the vinyl records that would be successful in these parties [20].

Digital samplers: The arrival of compact digital memory at the end of the 1970's made devices possible that allowed for quick sampling and manipulation of audio. Along with these digital (hardware) samplers came flexibility in control over the playback speed, equalisation and some other parameters such as the sample frequency. Signal processing power of hardware samples was initially limited compared to what software samplers can do nowadays. Classically, no time-stretching was provided in a way that didn't affect the frequency content of a sound. Samplers who did, produced audible artefacts that were desired in only very specific contexts. Two of the first widely available (and affordable) samplers were the *Ensoniq Mirage* (1985) and the *Akai S1000* (1989) [19]. An *Akai S1000* interface is shown with its keyboard version *Akai S1000 KB* in Figure 1.2.

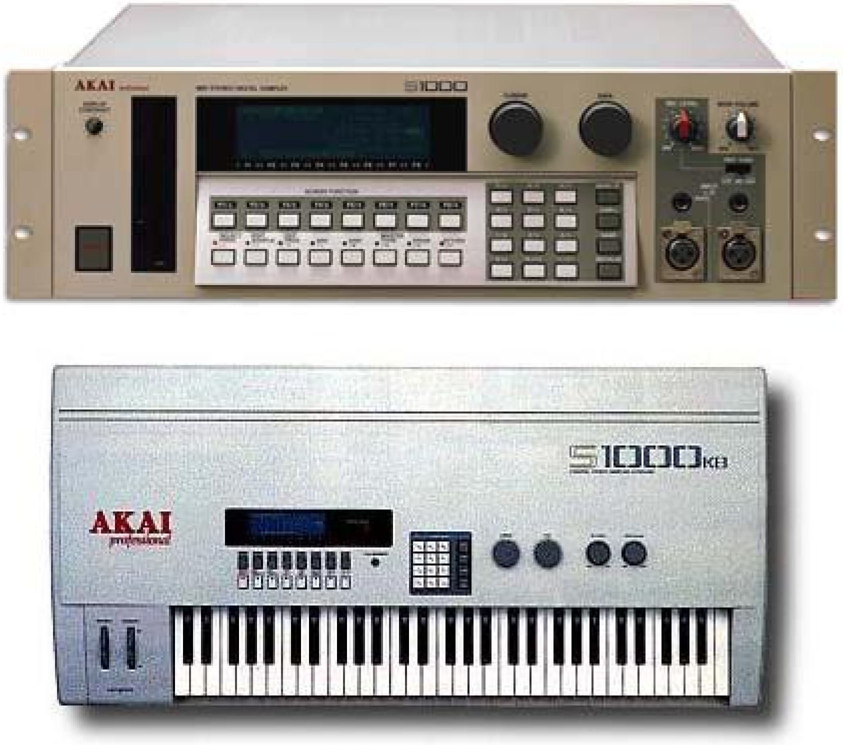
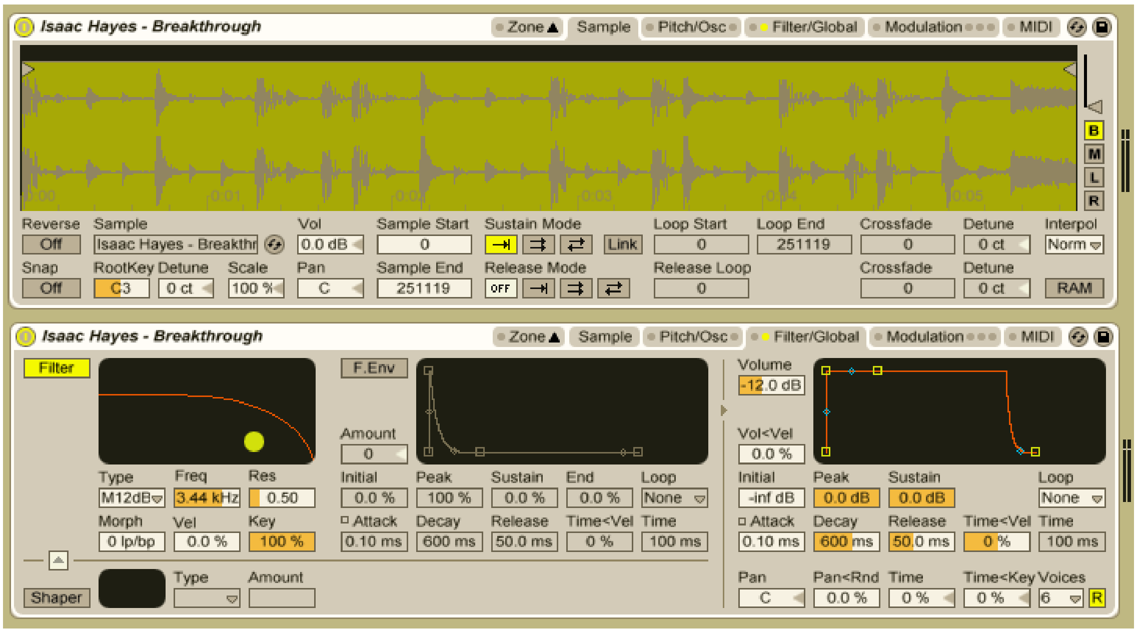


Figure 1.2: *Akai S1000* hardware sampler and its keyboard version *Akai S1000KB* (from www.vintagesynth.com).

Software samplers: The first powerful hardware samplers could in their days be seen as specialized audio computers, yet it didn't take long before comparable functionalities became available on home computers. Software samplers nowadays are generally integrated in digital audio workstations (DAW's) and provide independent transposition and time-stretching by default. A notable software sampler is Ableton's *Sampler* for Ableton's popular DAW *Live*, a screenshot is shown in Figure 1.3.



Reverse Off
Sample Isaac Hayes - Breakthr
Vol 0.0 dB
Sample Start 0
Sustain Mode
Link
Loop Start 0
Loop End 251119
Crossfade 0
Detune 0 ct
Interpol Norm
Snap Off
RootKey C3
Detune 0 ct
Scale 100 %
Pan C
Sample End 251119
Release Mode OFF
Release Loop 0
Crossfade 0
Detune 0 ct
RAM

Type M12dB
Freq 3.44 kHz
Res 0.50
Morph 0 lp/bp
Vel 0.0 %
Key 100 %
Amount 0
Initial 0.0 %
Peak 100 %
Sustain 0.0 %
End 0.0 %
Loop None
□ Attack 0.10 ms
Decay 600 ms
Release 50.0 ms
Time<Vel 0 %
Time 100 ms
Type
Amount
Volume -12.0 dB
Vol<Vel 0.0 %
Initial -inf dB
Peak 0.0 dB
Sustain 0.0 dB
Loop None
□ Attack 0.10 ms
Decay 600 ms
Release 50.0 ms
Time<Vel 0 %
Time 100 ms
Pan C
Pan<Rnd 0.0 %
Time 0 %
Time<Key 0 %
Voices 6
R

Figure 1.3: Screenshot of two panels of Ableton Live's *Sampler*. The panels show the waveform view and the filter parameters, amongst others. ©Ableton AG

1.2.3 Musical Content

In this section, the musical content of samples is described. This will be an important basis for the formulation of the requirements a sample recognition should meet. Note that no thorough musicological analysis could be found that lists all of the properties of samples relevant to the problem addressed in this thesis. Many of the properties listed in this section are therefore observations made when listening to many samples with their originals, rather than facts.

From this point in this thesis on, all statements on sampling refer to hip hop samples only, unless specified otherwise.

Origin

A large part of hip hop songs samples from what is sometimes referred to as African-American music, or in other cases labeled Rhythm&Blues, but almost all styles of music have been sampled, including classical music and jazz. Rock samples are less common than e.g. funk and soul samples, but have always been a significant minority. Producer Rick Rubin is known for sampling many rock songs in his works for Beastie Boys.

A typical misconception is that samples always involve drum loops. Vocal samples, rock riffs, brass harmonies, etc. are found just as easily and many samples feature a mixed instrumentation. In some cases, instrumentals or stems (partials tracks) are used. This being said, it is true that many of the first producers of rap music sampled mainly 'breaks'. A break in funk music is a short drum solo somewhere in the song, usually built on some variation of the main drum pattern [20]. Some record labels even released compilations of songs containing those breaks, such as the 'Ultimate Breaks and Beats' collection. This series of albums, released between 1986 and 1991 by Street Beat records, compiled popular and rare soul, funk and disco songs. It was released for DJ's and producers interested in sampling these drum grooves.[2]

After the first lawsuits involving alleged copyright infringements, many producers have chosen to rerecord their samples in a studio, in order to avoid fines or lengthy negotiations with the owners of the material. This kind of samples is referred to as 'interpolations'. The advantage for the producer is that he/she can keep the most interesting aspects of a sample, but deviate from it in others. Because of these possibly strong deviations, it is not the initial ambition of this work to include interpolations in the retrieval task.

Samples can also be taken from film dialogue or comedy shows. Examples are a sample from the film *The Mack* (1978) by Dr. Dre in *Rat Tat Tat Tat* (2001) and a sample taken from Eddie Murphy's comedy routine *Singers* (1987) in Public Enemy's *911 is a Joke* (1990, see also entry T153 in Appendix B). A radio play entitled *Frontier Psychiatrist* has been sampled in *Frontier Psychiatrist* (2000) by The Avalanches, a collective known for creating *Since I Left You* (2000), one of the most famous all-sample albums. In the context of this thesis, non-musical samples will not be studied.

Length

The length of samples varies from genre to genre and from artist to artist. In complex productions, samples can even be chopped up in very short parts, to be played back in a totally different order and combination. The jungle genre (a precursor of drum&bass) is the primary example of this [22]. It is often said that all early jungle tracks were built on one drum loop known as the Amen Break, sampled from The Winstons' *Amen Brother* (1969; see also entry T116 in Appendix B), but rearranged and played at a much faster tempo. The break would be the most frequently sampled piece of audio ever released, but this could not be verified. In hip hop, short samples appear as well. They can be as short as one drum stroke taken from an existing but uncredited record. Detecting very short samples obviously makes the identification more difficult, both for humans and automatic systems.

Recently in hip hop and R&B, the thin line between sampling and remixing has faded to the extent that large portions of widely known songs reappear almost unchanged. The Black Eyed Peas song *Pump It* mentioned earlier is an example. In other cases of long

[2] Note that the legal implications of sampling have remained uncertain until 1991, when rapper Biz Markie was the first hip hop artist to be found guilty of copyright violation. This was the famous *Grand Upright Music, Ltd. v. Warner Bros. Records Inc.* lawsuit about the sample of a piano riff by Gilbert O'Sullivan in Markie's song *Alone Again*) [21].

samples, the sampled artist might appear as a collaborator on the song, as is for example the case with Eminem ft. Dido's *Stan* (2000). It samples the chorus of Dido's *Thank You* (2000; see entries T063 and T062 in Appendix B).

Playback speed

Samples as they appear in popular music, hip hop and electronic music often differ from their original in the speed at which they are played back. This can change the perceived mood of a sample. In early hip-hop, for example, the majority of known samples were taken from soul or funk songs. Soul samples could be sped up to make them more danceable while funk songs could be slowed down to give rhythms a more laid back feel.

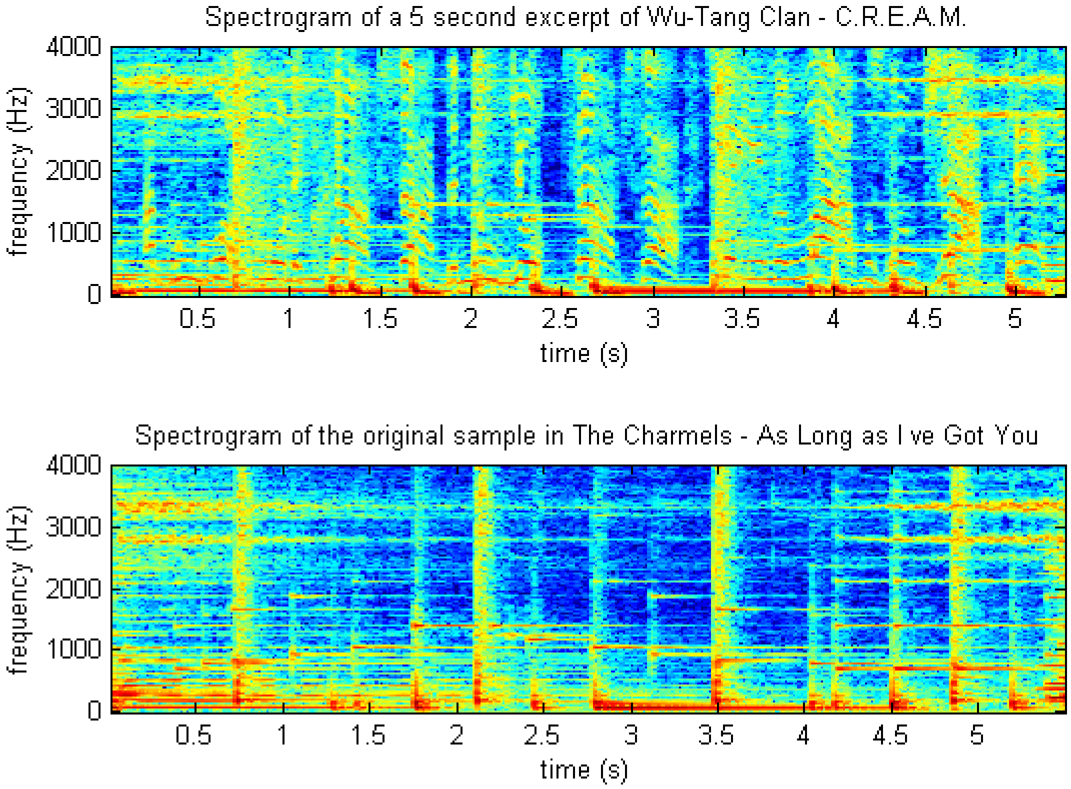
Usually, the sample is not the only musical element in the mix. To make tonal samples compatible with other instrumental layers, time-stretching can be done in way that does not affect the pitch, or is done by factors corresponding to discrete semitone repitches. For drums, inter-semitone pitch shifts are possible, provided there is no pitched audio left anywhere in the sample. Until recent breakthroughs around 1999 and 2003, time-stretching without pitch-shifting generally couldn't be done without some loss of audio quality [25, 26]. In most software samplers nowadays, this is easily accomplished.

In hip hop, repitches tend to be limited to a few semitones, with a small number of exceptions in which vocal samples are intended to sound peculiarly high pitched or drums to be drum&bass-like. Figure 1.4 shows the spectrogram of a 5 second sample (from Wu-Tang Clan - *C.R.E.A.M.*) and its original corresponding excerpt (from The Charmels - *As Long As I've Got You*). The bottom spectrogram reflects the presence of a simple drum pattern and some arch-shaped melody. The unsteady harmonics of the voice in the hip hop song (top), suggesting speech rather than singing, correspond to rap vocals indeed. Closer inspection of the frequencies and lengths reveals that the sample has been re-pitched one semitone up.

Filtering and Effects

The typically observed parameters controlling playback in samplers include filtering parameters, playback mode (mono, stereo, repeat, reverse, fade-out...) and level envelope controls (attack, decay, sustain, release). Filtering can be used by producers to maintain only the most interesting part of a sample. In drum loops, for example, a kick drum or hi-hat can be attenuated when a new kick or hi-hat will be added later. In almost all commercial music, compression will be applied at various stages in the production and mastering process.

Other more artistic effects that can be heard include reverberation and delay, a typical example being the very prominent echo effects frequently used in dub music [11], for example to mask the abrupt or unnatural ending of a sampled phrase. Naturally, each of these operations complicates the automatic recognition.



Spectrogram of a 5 second excerpt of Wu-Tang Clan - C.R.E.A.M.

Spectrogram of the original sample in The Charmels - As Long as I ve Got You

Figure 1.4: Spectrograms of a 5 second sample (top) and its original (bottom).

As a last note on the properties of samples, it is important to point out that a sample is generally not the only element in a mix. It appears between layers of other musical elements that complement it musically but, as a whole, are noise to any recognition system. Given that it is not unusual for two or more sample to appear at the same time, signal to noise ratios (SNR) for these samples can easily go below zero.

1.2.4 Creative Value

The creative value of the use of samples can be questioned and its debate is as old as the phenomenon itself. Depending as much on the author as on the case, examples of sampling have been characterized ranging from ‘obvious thievery’ (in the famous 1991 *Grand Upright Music, Ltd. v. Warner Bros. Records Inc.* lawsuit) to ‘the post-modernist artistic form par excellence’ [27].

Several scholars have placed sampling in a broader cultural context, relating it to traditional forms of creation and opposing it to the Western romantic ideal of novelty and the ‘autonomous creator’ [27, 21]. Hesmondhalgh states that “the conflict between Anglo-

American copyright law and sample-based rap music is obvious: the former protects what it calls 'original' works against unauthorized copying (among other activities), whereas the latter involves copying from another work to produce a 'derivative product". He then quotes Self, who concludes that this can indeed be seen as "a broader tension between two very different perspectives on creativity: a print culture that is based on ideals of individual autonomy, commodification and capitalism; and a folk culture that emphasizes integration, reclamation and contribution to an intertextual, intergenerational discourse" [8, 21]. Nevertheless has sampling become a wide-spread tool in many genres, and as even criticists admit, the sampler has become 'as common in the recording studio as the microphone' [28].

1.3 Research Outline

The goal of this thesis is to design and implement a automatic system that, given a hip hop song and a large music collection, can tell when the hip hop song samples any portion of the songs in the collection. Its definition may be simple, but to the best of the authors' knowledge, this problem has not been addressed before. Judging by the observed properties of samples and the current state-of-the-art in audio identification (see Chapter 2), the task is indeed very difficult. To illustrate this, and refine the goals, a first list of requirements for the sample recognition system can be stated.

1. Given a music collection, the system should be able to identify query audio that is known to the system, but heavily manipulated. These segments may be:
   - Very short,
   - Transposed,
   - Time-stretched,
   - Heavily filtered,
   - Non-tonal (i.e. purely percussive),
   - Processed with audio effects and/or
   - Appearing underneath a thick layer of other musical elements.
2. The system should be able to do this for large collections (e.g. over 1000 files).
3. The system should be able to do this in a reasonable amount of time (e.g. up to several hours).

The above requirements will be compared to those of audio fingerprinting and other music information retrieval systems in the next chapter. Some requirements are rather new to information retrieval tasks, the short length and possible non-tonal nature of samples being primary examples. Special attention will go to this non-tonality as well as transpositions and timestretches for reasons also explained in chapter 2.

1.3.1 Document Structure

Chapter 2 contains a review of the most relevant existing research in Music Information Retrieval. This includes some notes on frame-based audio processing and a general description of the audio identification problem. Details are also given for several existing types of audio identification systems, and their characteristics are critically discussed. As a last section, the chapter will include the detailed description of an implementation of one of these systems.

To evaluate the proposed systems, a music collection and an evaluation methodology are needed. Chapter 3 reports on the compilation of a representative dataset of sampling examples. This is an important part of the research and includes the manual annotation of a selection of relevant data. Chapter 3 also includes the selection of evaluation metrics that will be used, and the calculation of their random baselines.

In Chapter 4, a state-of-the-art audio identification system is optimised to obtain a state-of-the-art performance baseline for the sample recognition task. In Chapters 5 and 6, changes to the optimised approach are proposed to obtain a new system that fulfills as many of the above requirements possible. Each of the proposals is evaluated. Chapters 7 discusses the results of these evaluations and draws conclusions about what has been achieved. The conclusions lead to proposals for some possible future work.

Chapter 2

State-of-the-Art

2.1 Audio Representations

The following very short section touches on some concepts in frame-based audio analysis. Its purpose is not to introduce the reader to the general methodology, but to include some relevant definitions for reference and situate the most-used variables in this report.

Frame-based audio analysis is used here to refer to the analysis of audio in the time and frequency domain together. It requires cutting the signal into frames and taking of every frame a transform (e.g. Fourier) to obtain its (complex) spectrum. The length and overlap of the frames can vary depending on the desired time and frequency resolution.

2.1.1 Short Time Fourier Transform

The Discrete Fourier Transform

The discrete Fourier Transform (DFT) will be used to calculate the magnitude spectrum of signals. For a discrete signal $x(n)$ the DFT $X(f)$ is defined by

$$X(f) = \sum_{n=0}^{N-1} x(n)\, e^{-\frac{j2\pi f n}{N}}$$

where

- $n = 1 \ldots N$ is the discrete time variable (in samples)
- $f = 0 \ldots N$ are the discrete frequencies (in bins).
- N is the length of the signal $x(n)$.

The DFT is easily and quickly calculated with the Fast Fourier Transform (FFT) algorithm. Taking the magnitude $|X(f)|$ of $X(f)$ returns the magnitude spectrum and discards all phase information.

The Short Time Fourier Transform

The Short Time Fourier Transform (STFT) will be used to calculate the temporal evolution of the magnitude spectrum of signals. It is a series of DFT's of consecutive windowed signal portions.

$$X(f,t) = \sum_{n=0}^{N-1} w(n)\; x(Ht+n)\; e^{-\frac{j2\pi fn}{N}}$$

where t is the discrete time in frames. Important parameters are

- The window type used $w(n)$.
  In this thesis, a Hann window is used if nothing is specified.
- The window size N.
  The FFT size is assumed N or the next power of two is used unless specified.
- The hop size H.
  This variable is often defined by specification of the overlap factor $\frac{N-H}{N}$.

The magnitude yields the spectrogram of the function.

$$S(f,t) = |X(f,t)|$$

2.1.2 Constant Q Transform

A different approach to frequency analysis involves the Constant Q Transform (CQT) [29]. This transform calculates a spectrum in logarithmically spaced frequency bins. Such a spectrum representation with a constant number of bins per octave is more representative of the behaviour of the Human Auditory System (HAS) and the spacing of pitches in Western music [30, 29]. It was proposed by Brown in 1991 as [29]:

$$X(k) = \frac{1}{N_k} \sum_{n=0}^{N_k-1} w(n,k)\; x(n)\; e^{-\frac{j2\pi Qn}{N}}.$$

where

- the size N_k of the window $w(n,k)$ changes for every bin

- the (constant) Q is the 'quality factor'. It corresponds to the quality factor of an ideal filter bank that has the desired number of bands per octave:

$$Q = \frac{f_k}{\delta f_k}$$

  - f_k is the center frequency at bin k
  - δf_k the frequency difference to the next bin

Quality factor Q is kept constant in n and k, hence the logarithmically spaced central frequencies. For a resolution of 12 bins per octave (a semitone), Q takes a value around 17. A resolution of three bins per semitone requires a Q of approximately 51.

A fast algorithm to compute the constant Q transform has been proposed by Brown and Puckette [31]. It uses a set of kernels to map the output of a FFT to logarithmically spaced frequency bins. A version of this algorithm has been made available by Ellis[1] [32]. This implementation performs the mapping in the energy (squared magnitude) domain, decreasing computation time at the expense of losing phase information. It also allows the user to specify the used FFT size. Putting constraints on the FFT sizes result in a blurring of the lowest frequencies, but an increase in efficiency.

The implementation has the following parameters:

- The FFT size N in ms (as with the STFT).
- The hop size H in ms (as with in the STFT).
- The central frequency f_{min} of the lowest bin $k = 0$.
- The sample rate SR determining the highest frequency $f_{max} = SR/2$.
- The number of bins per octave *bpo* determining Q as follows:

$$Q = 2^{1/bpo} - 1.$$

The algorithm returns a matrix with columns of length K, where K is the number of resulting logarithmically spaced frequency bins as determined by f_{min}, f_{max} and *bpo*.

2.2 Scientific Background

The problem of sample identification can be classified as an audio recognition problem applied to short or very short music fragments. In this sense, it faces many of the challenges that are dealt with in audio fingerprinting research. The term audio fingerprinting is used

[1] http://www.ee.columbia.edu/ dpwe/resources/matlab/sgram/

for systems that attempt to identify unlabeled audio by matching a compact, content-based representation of it, the fingerprint, against a database of labeled fingerprints [2].

Just like sample recognition systems, fingerprinting systems are often designed to be robust to noise and several transformations such as filtering, acoustic transmission and GSM compression in cell phones. However, in the case of samples, the analysed audio can also be pitch-shifted or time-stretched and it can contain several layers of extra instruments and vocals, etc. (as described in Chapter 1). Because of this unpredictable appearance, the problem of sample identification also relates to cover detection [16]. Cover detection or version identification systems try to assess if two musical recordings are different renditions of the same musical piece. In state of the art cover detection systems, transpositions and changes in tempo are taken into account.

Then again, the context of sampling is more restrictive than that of covers. Even though musical elements such as melody or harmony of a song are generally not conserved, low-level audio features such as timbre aspects, local tempo, or spectral details could be somehow invariant under sampling. Thus, the problem can be situated between audio fingerprinting and cover detection and seems therefore related to recognition of remixes [33]. It must however be mentioned that 'remix' is very broad term. It is used and understood in many ways, and not all of those are relevant (e.g. the literal meaning of remix).

Sample detection shares most properties with remix detection. To show this, one could attempt to make a table listing invariance properties for the three music retrieval tasks mentioned, but any such table depends on the way the tasks are understood. Moreover, both for remix recognition and cover detection it has been pointed out that basically any aspect of the song can undergo a change. The statement that sample detection relates most to remix detection is therefore based on the observation that remixes, as defined in [33], are de facto a form of sampling as it has been defined in Chapter 1. The next section is an overview of said research on remix recognition.

2.3 Remix Recognition

The goal in remix recognition is to detect if a musical recording is a remix of another recording known to the system. The problem as such has been defined by Casey and Slaney [33].

The challenge in recognizing remixed audio is that remixes often contain only a fraction of the original musical content of a song. However, very often this fraction includes the vocal track. This allows for retrieval through the matching of extracted melodies. Rather, though, than extracting these melodies entirely and computing melodic similarities, distances are computed on a shorter time scale. One reason is that, as researchers in cover detection have pointed out, melody extraction is not reliable enough (yet) to form the basis of a powerful music retrieval system [16].

2.3.1 Audio Shingles

Casey et al. used ‘shingles’ to compute a ‘remix distance’ [33]. Audio shingles are the audio equivalent of the text singles used to identify duplicate web pages. Here, word histograms are extracted for different portions of the document. These histograms can then be matched against a database of histograms to determine how many of the examined portions are known to the system. Audio shingles work in a comparable way.

Shingles

The proposed shingles are time series of extracted features for 4 seconds of windowed audio. They are represented by a high-dimensional vector. The remix distance d between two songs A and B is then computed as the average distance between the N closest matching shingles. It can formally be defined as

$$d(A,B) = \sum_{N} min_{i,j}^{N} \sum_{k} \left| x_k^i - y_k^j \right|^2 ,$$

with $x^i \in A$ and $y^j \in B$, shingle vectors drawn for the songs i and j.

The features used by the authors are PCP’s and LFCC’s, computed every 100ms. PCP’s (pitch class profiles) are 12 dimensional profiles of the frequencies present in the audio, where the integrated frequencies span multiple octaves but are collapsed into semitone partitions of a single octave. LFCC’s (Logarithmic Frequency Cepstrum Coefficients) are a 20-dimensional cepstrum representation of the spectral envelope. Contrary to MFCC’s the features used here are computed in logarithmically spaced bands, the same 12th octave bands as used when computing the PCP’s.

Figure 2.1 shows a block diagram of the shingle extraction. To combine the features into shingles, the audio must be sliced to windows, and then to smaller frames by computing the STFT (short time fourier transform)[2]. For implementation details regarding PCP and LFCC’s, refer to [33]. The result of the extraction is a set of two descriptor time series for every 4s window, in the form of two vectors of very high dimension: 480 and 800 respectively. An important (earlier) contribution of the authors is to show that Euclidian distances in these high-dimensional spaces make sense as a measure of musical similarity, and that ‘the curse of dimensionality’ is effectively overcome [34].

Locality Sensitive Hashing

Identifying neighbouring shingles in such high dimensional spaces is computationally expensive. To quickly retrieve shingles close to a query, i.e. less than a certain threshold r

[2] Note that, as can be seen in the diagram, the 4 s windows and STFT frames have the same hop size (100 ms). In practice therefore, the STFT can be computed first and the windows can be composed by simply grouping frames.

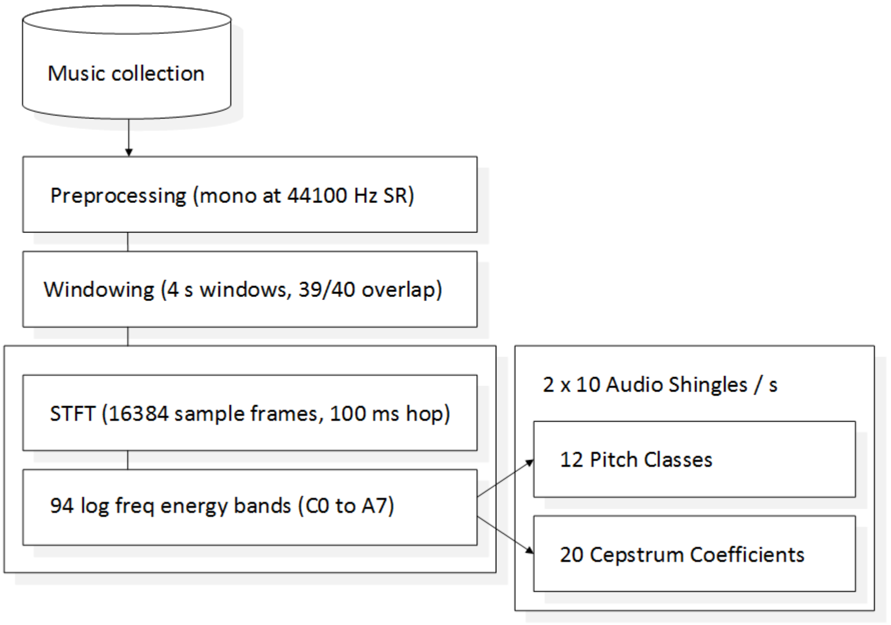

Music collection
Preprocessing (mono at 44100 Hz SR)
Windowing (4 s windows, 39/40 overlap)
STFT (16384 sample frames, 100 ms hop)
94 log freq energy bands (C0 to A7)
2 x 10 Audio Shingles / s
12 Pitch Classes
20 Cepstrum Coefficients

Figure 2.1: Simplified block diagram of the extraction of audio shingles.

away, the described system uses a hashing procedure known as Locality Sensitive Hashing (LSH). Generally in LSH, similar shingles are assigned neighbouring hashes, whereas normal hashing will assign radically different hashes to similar items, so as only to allow retrieval of items that are exactly identical.

The authors compute the shingles' hashes by projecting the vectors x_i on a random one-dimensional basis V. The real line V is then divided into equal parts, with a length corresponding to the similarity threshold r. Finally, the hash is determined by the index of the part to which the vectors are projected. In a query, all shingles with the same hash as the query are initially retrieved, but only those effectively closer than r are kept after computing the distances. Figure 2.2 shows a histogram of how many shingles are retrieved for relevant and non-relevant tracks in a remix recognition task.

Discussion

The overall performance of this method is reported to be good. In [1], the researchers use the same algorithm to perform three tasks: fingerprinting, cover detection and remix recognition. Precision and recall are high, suggesting that the algorithm could be success-

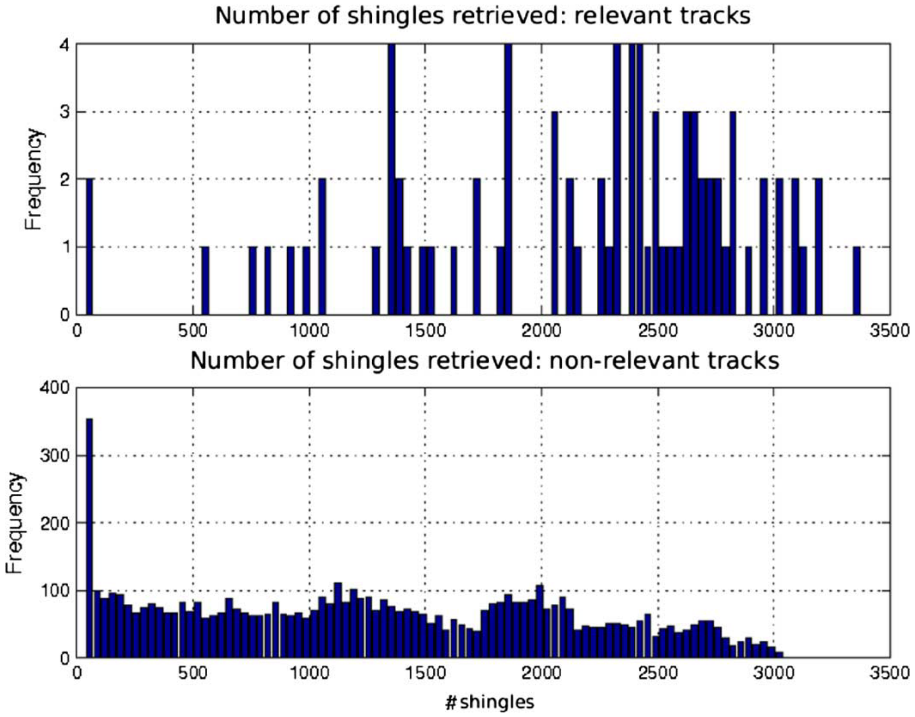


Figure 2.2: Histogram of retrieved shingle counts for the remix recognition task [1]. The upper graph shows the counts for relevant data and the lower shows counts for non relevant data. A high number of shingles means a high similarity to the query (and therefore a small distance).

ful in the recognition of samples. However, some comments need to be made.

The evaluation is limited to carefully selected tasks. For example, in the case of cover detection the system is used to retrieve renditions of a classical composition (a Mazurka by Chopin). The use of Chopin Mazurkas in Music Information Retrieval is popular, but its use in the evaluation of Cover Detection algorithms has been criticized [35]. It is clear that all performances of this work share the same instrumentation. In addition, the key in which it is played will very likely not vary either. Contrary to what is suggested in the author's definition of remix detection in [33], the system as it is described does indeed not account for any major pitch or key variations, such as a transposition (nor changes in instrumentation, structure and global tempo).

The tasks of sample identification and remix recognition are similar, but not the same. Transpositions will generally occur more often in sampled music than in remixes. Second and more important, remix recognition is said to rely on detecting similarity of the predominant musical elements of two songs. In the case of sampling, the assumption that the predominant elements of sample and original correspond, is generally wrong. The LFCC features used to describe the spectrum will not be invariant to the addition of other musical layers. Finally, using Pitch Class Profiles would assume not only predominance of the sample, but also tonality. As said earlier, this is often not the case.

In extension of this short review, one could say that these last arguments do not only go for the work by Casey, but also for other research in audio matching such as by Kurth and Müller [36], and in extent for all of cover detection: matching tends to rely largely on predominant musical elements of two songs and/or tonal information (in a minority of cases timbral information) [16]. For sample recognition, this is not an interesting starting point. However, many things could nevertheless be learned from other aspects of audio matching, such as how to deal with transpositions.

2.4 Audio Fingerprinting

Audio fingerprinting systems make use of audio fingerprints to represent audio objects for comparison. An audio fingerprint is a compact, perceptual digest from a raw audio signal that can be stored in a database so that pairs of tracks can be identified as being the same. A very widespread implementation for audio identification is the Shazam service, launched in 2002 and available for iPhone shortly after its release [37].

A comprehensive overview of early fingerprinting techniques (including distances and searching methods) is given by Cano et al. [2]. It lists the main requirements that a good system should meet and describes the structure and building blocks of a generalized content-based audio identification framework. Around the same time, there were three systems being developed that will be discussed subsequently.

The work that is reviewed in most detail here relates to fingerprinting and is already over eight years old. This is because the problem of robust audio identification can be regarded as largely solved by 2003, later related research expanded over audio similarity (rather than identity) to version detection and were situated in the chroma-domain [36].

2.4.1 Properties of Fingerprinting Systems

Requirements

There are three main requirements for a typical content-based audio identification system.

1. **Discriminative power**:
   The representation should contain enough information (or entropy) to discriminate over large numbers of other fingerprints from a short query.
2. **Efficiency**:
   The discriminative power is only relevant if this huge collection of fingerprints can be queried in a reasonable amount of time. The importance of the computational cost of the fingerprint extraction is decreasing as machines become more and more

powerful, yet the extraction of the fingerprint is still preferable done somewhere near real-time.

3. **Robustness**:
   The system should be able to identify audio that contains noise and/or has undergone some transformations. The amount and types of noise and transformations considered always depend on the goals set by the author.

The noise and distortions to be dealt with have ranged from changes in amplitude, dynamics and equalisation, DA/AD conversion, perceptual coding and analog and digital noise at limited SNR's [5, 30], over small deviations in tape and CD playback speed [38] to artifacts typical for poorly captured radio recordings transmitted over a mobile phone connection [6, 3]. The latter includes FM/AM transmission, acoustical transmission, GSM transmission, frequency loss in speaker and microphone and background noise and reverberation present at the time of recording.

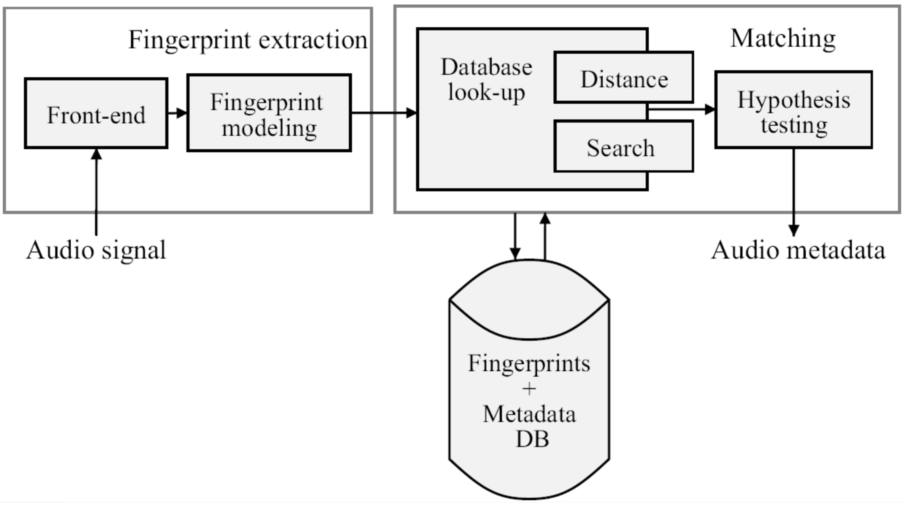


Figure 2.3: Block diagram of a generalized audio identification system [2].

Typical structure

A typical framework for audio identification will have an extraction and a matching block, as can be seen in Figure 2.3. Figure 2.4 shows a more detailed diagram of such an extraction block. It will typically include some pre- and postprocessing of the audio (features). Common preprocessing operations are mono conversion, normalisation, downsampling, and band-filtering to approximate the expected equalisation of the query sample. Possible postprocessing operations include normalisation, differentiation of obtained time series and low resolution quantisation.

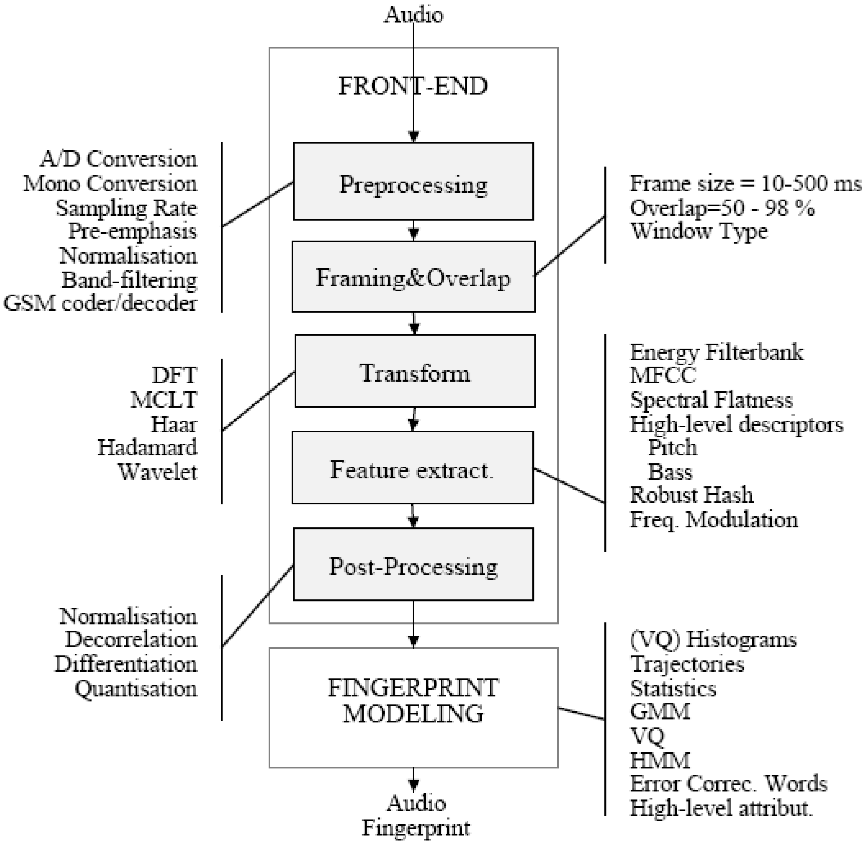


Figure 2.4: Diagram of the extraction block of a generalized audio identification system [2].

The efficiency of fingerprinting systems largely rely on their look-up method, i.e. the matching block. However, the many different techniques for matching will not be discussed in detail. As opposed to classical fingerprinting research, there is no emphasis on speed in this investigation, and it is the conviction of the authors that, first of all, accurate retrieval needs to be achieved. The following paragraphs review the most relevant previous research, focusing on the types of fingerprint used and their extraction.

2.4.2 Spectral Flatness Measure

In 2001, Herre et al. presented a system that makes use of the spectral flatness measure (SFM) [5]. The paper is not the first to research content-based audio identification but it is one of the first to aim at robustness. The authors first list a number of features previously used in the description and analysis of audio and claim that there are no natural candidates amongst them that provide invariance to alterations in both absolute signal level and coarse spectral shape. The arguments are summarized in Table 2.1.

Energy Loudness	Depend on absolute level
Band-width Sharpness Brightness Spectral centroid Zero crossing rate	Depend on coarse spectral shape
Pitch	Only applicable to a limited class of audio signals

Table 2.1: List of traditional features that, according to [5], cannot provide invariance to both absolute signal level and coarse spectral shape.

Methodology

Herre et al. then show that the spectral flatness measure provides the required robustness and so does the spectral crest factor (SCF). The SFM and SCF are computed per frequency band k containing the frequencies $f = 0 \ldots N-1$.

$$SFM_k = \frac{\left[\prod_f S_k^2(f)\right]^{\frac{1}{N}}}{\frac{1}{N}\sum_f S_k^2(f)}$$

$$SCF_k = \frac{\max_k S_k^2(f)}{\frac{1}{N}\sum_k S_k^2(f)},$$

where S_k^2 is the power spectral density function in the band[3] [4].

Both measures are calculated and compared in a number of different frequency bands (between 300 and $6000Hz$). The perceptual equivalent of these measures can be described as noise-likeness and tone-likeness. In general, features with perceptual meaning are assumed to represent characteristics of the sound that are more likely to be preserved and should thus promise better robustness.

Only few details about the matching stage are given by the authors. The actual fingerprints consist of vector quantization (VQ) codebooks trained with the extracted feature vectors. Incoming feature vectors are then quantized using these codebooks. Finally, the database item that minimizes the accumulated quantization error is returned as the best match.

Evaluation and Discussion

Evaluation of this approach is done by matching distorted queries against a database of 1000 to 30000 items. All SFM related results for two of the distortion types are

[3] Recall that in this thesis, S denotes the magnitude spectrum, while X is the complex spectrum.

[4] Generally N depends on k, but this N_k is simplified to N for easy notation.

Distortion type	Window	Bands	Band spacing	Set size	Performance
cropped MP3 @ 96kbit/s	1024	4	equal	1000	90.0%
cropped MP3 @ 96kbit/s	1323	4	equal	1000	94.6%
cropped MP3 @ 96kbit/s	1323	16	equal	1000	94.3%
cropped MP3 @ 96kbit/s	1323	16	logarithmic	30000	99.9%
cheap speakers and mic	1024	4	equal	1000	27.2%
cheap speakers and mic	1323	4	equal	1000	45.4%
cheap speakers and mic	1323	16	equal	1000	97.5%
cheap speakers and mic	1323	16	logarithmic	30000	99.8%

Table 2.2: A selection of experiments illustrating the performance of the SFM-based fingerprinting system with experimental setup details as provided in [5].

given in Table 2.2 as a summary of the reported performance (results for the SCF were not significantly different). Window sizes are expressed in samples, the performance is expressed as the number of items that were correctly identified by the best match. It is also mentioned in [5] that the matching algorithm has been enhanced between experiments but no details are given.

The reported performance is clearly good, almost perfect. The only conclusion drawn from these results is indeed that 'the features provide excellent matching performance both with respect to discrimination and robustness'. However, no conclusions can be made about which of the modified parameters accounts most for the improvement between experiments: the change from 4 to 16 bands, the logarithmic spacing of bands, or the change in the matching algorithm. More experiments would need to be done.

A secondary comment that can be made is that no information is given about the size of the representations. Fingerprint size and computation time may not be the most important attributes of a system that emphasises on robustness, yet with total absence of such information it cannot be told at what cost the performance has been taken to the reported percentages. Nevertheless, the authors show that the SFM and SCF can be successfully used in content-based audio identification.

2.4.3 Band energies

Herre et al. claimed that energy cannot be used for efficient audio characterization. However, their approach was rather traditional, in the sense that the extraction of the investigated features has been implemented without any sophisticated pre- or postprocessing. Haitsma et al. [30] present an audio fingerprint based on quantized energy changes across the two-dimensional time-frequency space. It is based on strategies for image fingerprinting.

Methodology

The system they present cuts the audio in windowed 400 ms frames (with overlap factor 31/32) and calculates in every frame the DFT. The frequencies between 300 and $3000 Hz$ are then divided into 33 bands and the energy is computed for every band. To stay true to the behaviour of the HAS, the bands are logarithmically spaced and non-overlapping. If time is expressed as the frame number t and frequency as the band number k, the result is a two-dimensional time-frequency function $E(t, k)$.

Of this $E(t, k)$, the difference function is taken in both the time and frequency domain, and quantized to one bit. This is done at once as follows:

$$\delta E(t,k) = \begin{cases} 1 & E(t,k) - E(t,k+1) - (E(t-1,k) - E(t-1,k+1)) > 0 \\ 0 & E(t,k) - E(t,k+1) - (E(t-1,k) - E(t-1,k+1)) \leq 0 \end{cases}$$

This results in a string of 32 bits for every frame T, called a subfingerprint or hash. The combination of differentiation and one bit quantisation provides some tolerance towards variations in level (e.g. from dynamic range compression with slow response) and smooth deviations of the coarse spectral shape (e.g. from equalisation with low Q).

Matching, roughly summarized, is done by comparing extracted bit strings to a database. The database contains bit strings that refer to song ID's and time stamps. If matching bit strings refer to consistent extraction times within the same song, that song is retrieved as a match. It is shown that a few matches per second (less then 5% of bit strings) should suffice to identify a 3 second query in a large database. To boost hits, probable deviations from the extracted subfingerprints can be included in the query. This is a way of providing some tolerance in the hashing system, though very likely at the cost of discriminative power.

Evaluation and Discussion

There is no report found on any evaluation of this exact system using an extended song collection and a set of queries. As a consequence, no conclusions can be made about the system's discriminative power in a real-life conditions. Instead, [6] studies subfingerprints extracted from several types of distorted 3 second queries, to study the robustness of the system. The effect of the distortions is quantified in terms of hits, i.e. hashes that are free of bit errors when compared to those of the original sound. Four songs of different genres and 19 types of distortion are studied. The types of distortion include different levels of perceptual coding, GSM coding, filtering, time scaling and the addition of white noise.

The results are summarized in Table 2.3. The signal degradations, listed in the rows, are applied to four 3 second songs excerpts, listed in the columns. The first number in every cell indicates the hits out of 256 extracted subfingerprints. The second number indicates

Distortion type	Carl Orff	Sinead O'Connor	Texas	AC/DC
MP3@128Kbps	17, 170	20, 196	23, 182	19, 144
MP3@32Kbps	0, 34	10, 153	13, 148	5, 61
Real@20Kbps	2, 7	7, 110	2, 67	1, 41
GSM	1, 57	2, 95	1, 60	0, 31
GSM C/I = 4dB	0, 3	0, 12	0, 1	0, 3
All-pass filtering	157, 240	158, 256	146, 256	106, 219
Amp. Compr.	55, 191	59, 183	16, 73	44, 146
Equalization	55, 203	71, 227	34, 172	42, 148
Echo Addition	2, 36	12, 69	15, 69	4, 52
Band Pass Filter	123, 225	118, 253	117, 255	80, 214
Time Scale +4%	6, 55	7, 68	16, 70	6, 36
Time Scale 4%	17, 60	22, 77	23, 62	16, 44
Linear Speed +1%	3, 29	18, 170	3, 82	1, 16
Linear Speed -1%	0, 7	5, 88	0, 7	0, 8
Linear Speed +4%	0, 0	0, 0	0, 0	0, 1
Linear Speed -4%	0, 0	0, 0	0, 0	0, 0
Noise Addition	190, 256	178, 255	179, 256	114, 225
Resampling	255, 256	255, 256	254, 256	254, 256
D/A + A/D	15, 149	38, 229	13, 114	31, 145

Table 2.3: Number of error-free hashes for different kinds of signal degradations applied to four songs excerpts. The first number indicates the hits for using only the 256 subfingerprints as a query. The second number indicates hits when the 1024 most probable deviations from the subfingerprints are also used. From [6].

hits when the 1024 most probable deviations from those 256 subfingerprints are also used as a query.

Theoretically, one matching hash is sufficient for a correct identification, but several matches are better for discriminative power. With this criterion, it becomes apparent that the algorithm is fairly robust, especially for filtering and compression. Distortion types that cause problems are GSM and perceptual coding, the type that causes the least trouble is resampling. However, there is enough information to conclude that this system would fail in aspects crucial to sample identification: speed changes and addition of effects.

First, even though the system handles changes made to the tempo quite well, experiments with changes in linear speed (tempo and pitch change together) do bad: none of the hashes are preserved. Second, the only experiment performed with the addition of noise uses white noise. The noise is constant in time and uniform in spectrum and poses as such no challenge to the system. Other types of noise (such as a pitched voice) are not tested but can be expected to cause more problems.

2.4.4 Landmarks

The most widely known implementation of audio fingerprinting has been designed by Wang and Smith for Shazam Entertainment Ltd., a London based company[5]. Their approach has been patented [39] and published [3]. The system is the first one to make use of spectral peak locations.

Motivation

Spectral peaks have the interesting characteristic of showing approximate linear 'superposability'. Summing a sound with another tends to preserve the majority of the original sound's peaks [39]. Spectral peak locations also show a fair invariance to equalization. The transfer functions of many filters (including acoustic transmission) are smooth enough to preserve spectral details on the order of a few frequency bins. If in an exceptional case the transfer function's derivative is high, peaks can be slightly shifted, yet only in the regions close to the cut-off frequencies [3].

Methodology

The general structure of the system is very comparable to the generalized framework described in section 2.4.1. An overview is given in Figure 2.5.

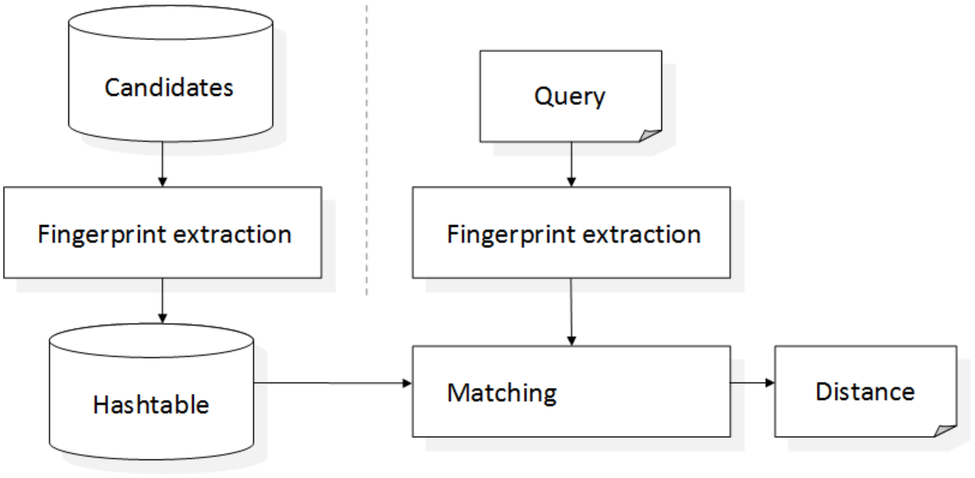


Figure 2.5: Block diagram overview of the landmark fingerprinting system as proposed by Wang [3].

[5] http://www.shazam.com/music/web/about.html

The extraction of the fingerprint is now explained. Rather than storing sets of spectral peak locations and time values directly to a database, Wang bases the fingerprint on 'landmarks'. Landmarks combine peaks into pairs of peaks. Every pair is then uniquely identified by two time values and two frequency values. These values can be combined in one identifier, which allows for faster look-up in the matching stage. The algorithm can be outlined as follows[6]:

Algorithm 2.1

1. Preprocess audio (no details are given).
2. Take the STFT to obtain the spectrogram $S(t, f)$.
3. Make a uniform selection of spectral peaks ('constellation').
4. Combine nearby peaks (t_1, f_1) and (t_2, f_2) into a pair or 'landmark' L.
5. Combine f_1 , f_2 and $\Delta t = t_2 - t_1$ into a 32-bit hash h.
6. Combine t_1 and the song's numeric identifier into a 32-bit unsigned integer ID.
7. Store ID in the database hash table at index h.

Just like the hashes in the energy-based fingerprinting system (section 2.4.3), the hashes obtained here can be seen as subfingerprints. A song is not reduced to just one hash, rather it is represented by a number of hashes every second. An example of a peak constellation and landmark are shown in Figure 2.6.

In the matching step, the matching hashes are associated with their time offsets t_1 for both query and candidates. For a true match between to songs, the query and candidate time stamps have a common offset for all corresponding hashes. Number of subfingerprint matching this way is computed as follows:

Algorithm 2.2

1. Extract all the query file's hashes $\{h\}$ as described in Algorithm 2.1.
2. Retrieve all hashes $\{h_d\}$ matching the query's set of hashes $\{h\}$ from the database, with their song id's $\{C_d\}$ and timestamps $\{t_{1d}\}$.
3. For each song $\{C_d\}$ referenced in $\{h_d\}$, compute the differences $\{t_{1d} - t_1\}$.

[6] Many details of the algorithm, such as implementation guidelines or parameter defaults, have not been published

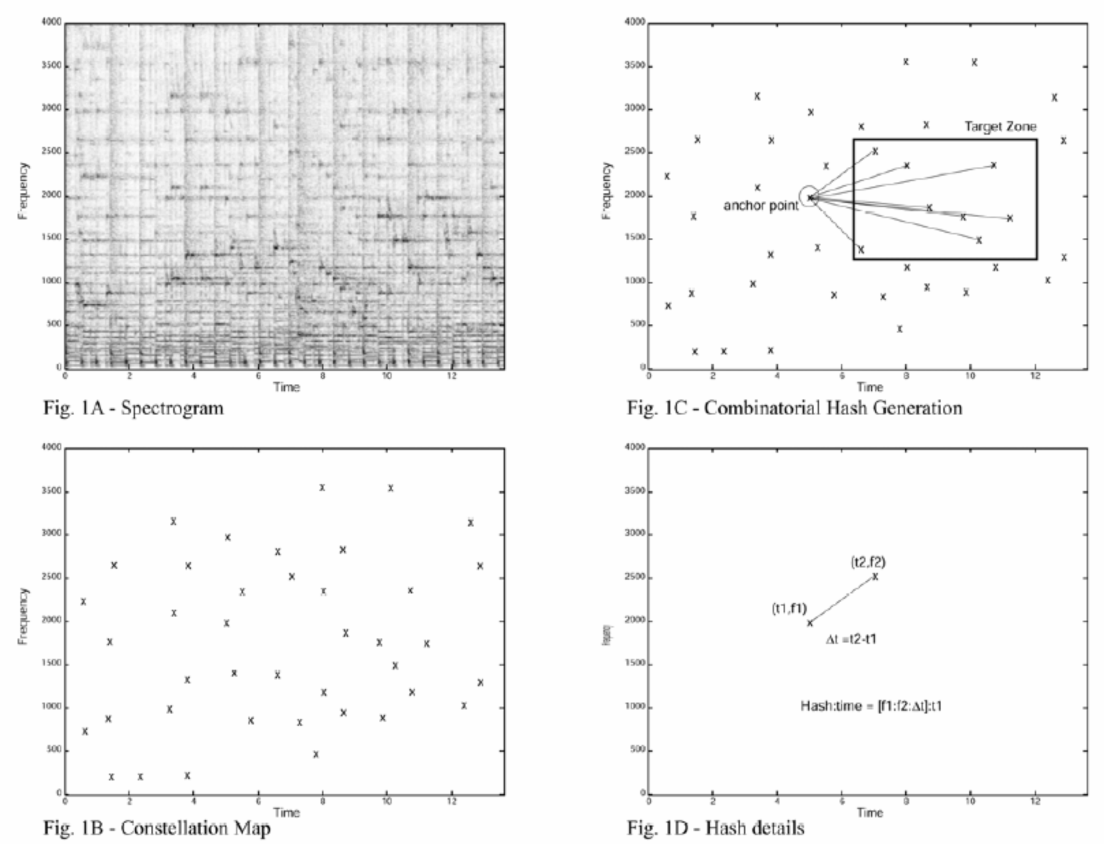


Figure 2.6: Reduction of a spectrogram to a peak constellation (left) and pairing (right). [3]

4. If a significant amount of the time differences for a song C_d are the same, there is a match.

The last matching step is illustrated in Figure 2.7 showing histograms of the time differences $\{t_{1d} - t_1\}$.

Landmarks can be visualised in a spectrogram. An example of a fingerprint constellation for an audio excerpt is given in Figure 2.8. The fingerprints are plotted as lines on the spectrogram of the analysed sound. Note that the actual number of landmarks and number of pairs per peak depends on how many peaks are found and how far they are apart.

Evaluation and discussion

The system is known to perform very well. A thorough test of the system is done in [3] using realistic distortions: GSM compression (which includes a lot of frequency loss) and

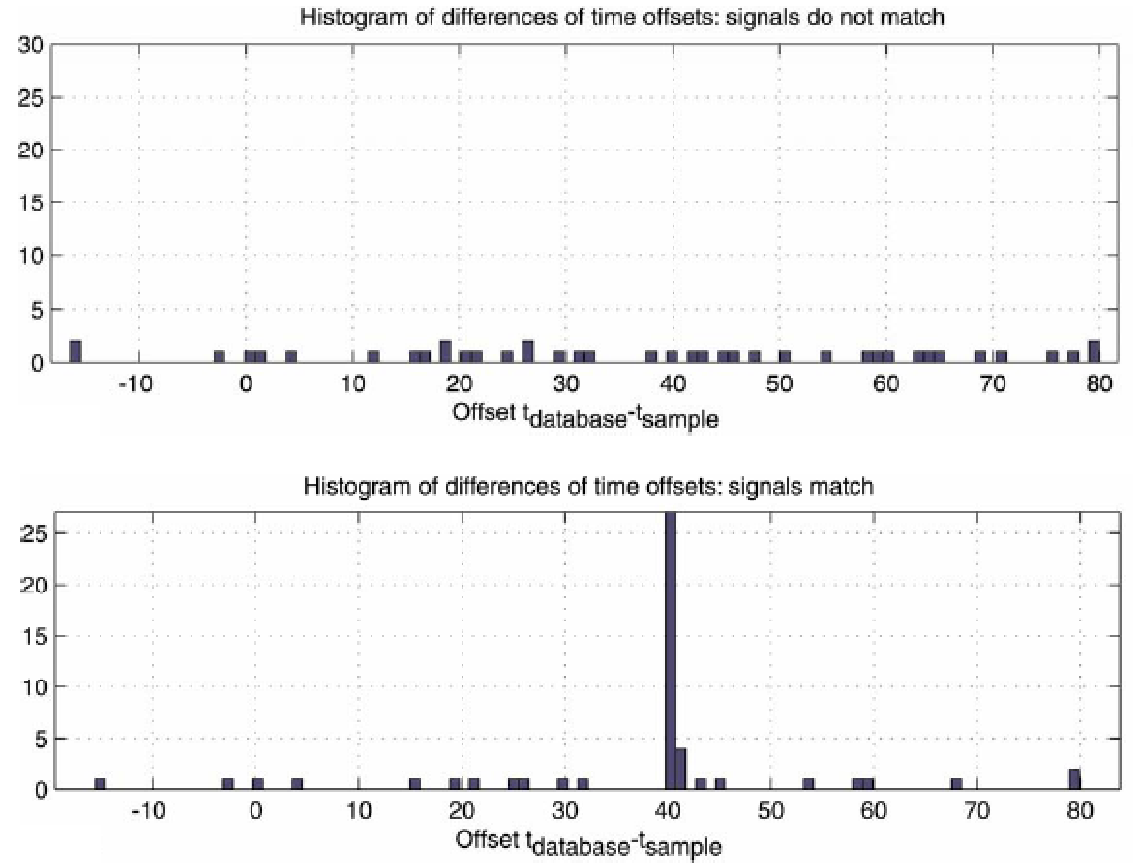


Figure 2.7: The time differences $t_d - t_1$ for non-matching tracks have a uniform distribution (top). For matching tracks, the time differences show a clear peak (bottom) [3].

addition of background noise recorded in a pub. The results show that high recognition rates are obtained even for heavily distorted queries, see Figure 2.9. It is also shown that only 1 or 2 % peaks survival is required for a match. Account of some of the experiences of Shazam in the commercialization of this invention confirms this.

Some advantages and disadvantages of spectral peak-based fingerprints in the context of sample identification are listed in Table 2.4. Clearly the algorithm has not been designed to detect transposed or time-stretched audio. However, the system is promising in terms of robustness to noise and transformations. An important unanswered question is if percussive sounds can be reliably represented in a spectral peak-based fingerprint. It can be noted that the proposed system has been designed to identify tonal content in a noisy context, and fingerprinting drum samples requires quite the opposite.

Two more remarks by Wang are worth including. The first one is a comment on a property the author calls ‘transparency’. He reports that, even with a large database, the system could correctly identify each of several tracks mixed together, including multiple versions

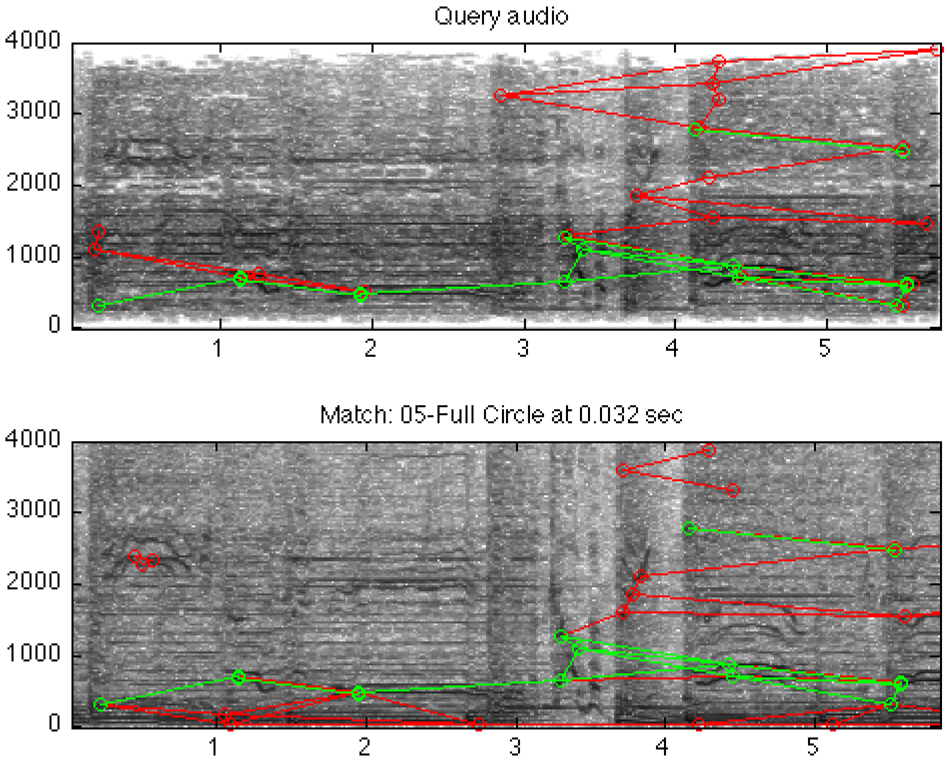


Figure 2.8: Fingerprints extracted from a query segment and its matching database file. Red lines are non-matching landmarks, green landmarks match. [4]

of the same piece. This is an interesting property that a sampling identification system ideally should possess. The second remark refers to sampling. Wang accounts:

> “We occasionally get reports of false positives. Often times we find that the algorithm was not actually wrong since it had picked up an example of ‘sampling,’ or plagiarism.”

2.4.5 Implementation of the Landmark-based System

An implementation of the described algorithm has been made by by Ellis [4]. The script is designed to extract, store, match and visualise landmark-based fingerprints as they have been originally conceived by Wang and is freely available on Ellis’ website[7].

[7] http://labrosa.ee.columbia.edu/matlab/fingerprint/

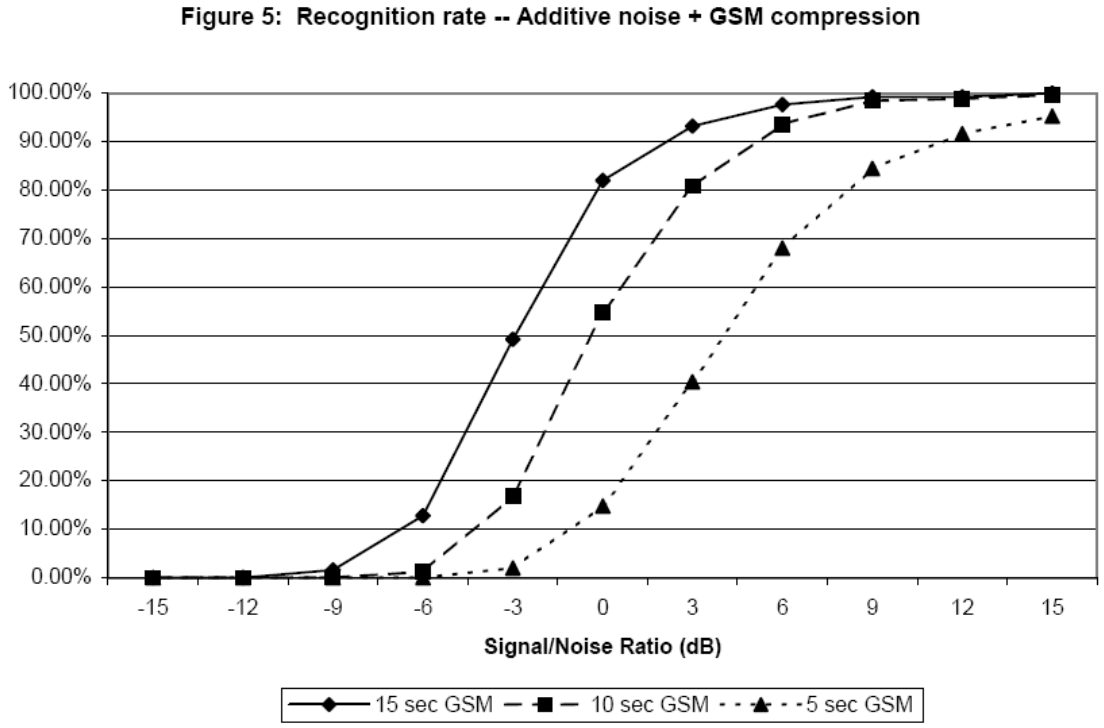


Figure 2.9: Evaluation results of the landmark fingerprinting system [3].

Overview

An overview of the proposed implementation is given as a block diagram in Figure 2.10. This is indeed a more detailed version of the diagram in Figure 2.5. A more detailed diagram of the separation of extraction components is given in Figures 2.11 and 2.12.

Important properties of this implementation are (details in the upcoming paragraphs):

- A **spectral peak** is defined as a local maximum in the log magnitude spectrum. The magnitude of a peak is higher than that of its neighbouring frequencies.
- A **uniform selection** of spectral peaks is made by selecting only those that exceed a masking curve that is incrementally updated with every peak found. The procedure is governed by many parameters.
- Absence of **hypothesis testing**: no criterion is implemented to decide if a match to a query is found. Instead, when the system is given a query, it returns the number of matching landmarks for every database file.

Advantages	Disadvantages
High robustness to noise and distortions.	Not suited for transposed or time-stretched audio.
Ability to identify music from only a very short segment.	Designed to identify tonal content in a noisy context, fingerprinting drum samples requires the opposite.
Does not explicitly require tonal content.	Can percussive recordings be represented by just spectral peaks at all?

Table 2.4: Advantages and disadvantages of spectral peak-based fingerprints in the context of sample identification.

Extraction

The extraction stage's algorithm description (repeated here in emphasized type) can now be supplemented with details about the implementation of every step. An overview of parameters is included with defaults in parentheses.

Algorithm 2.3

1. *Preprocess audio.*

   (a) Convert signal $x(n)$ to mono

   (b) Resample to samplerate SR (8000 Hz)

2. *Take the STFT to obtain the spectrogram $S(t, f)$.*
   STFT parameters are

   - The window type (Hann)
   - The window size N (64 ms)
   - Hop size H (32 ms)

   Further processing consists of

   (a) Taking magnitudes of spectra, discarding phase information.

   (b) Taking the logarithm of all magnitudes $S(t, f)$.

   (c) Apply a high-pass filter (HPF) to the spectrum curve as if it were a signal[8]. The filter has one control parameter *pole* (0.98), the positive pole of the HPF.

   These lasts steps are undertaken to make S less dependent on absolute level and coarse spectral shape.

[8] Before filtering, the mean of the spectrum is subtracted in every frame to minimize ripple in the low and high ends of the spectrum.

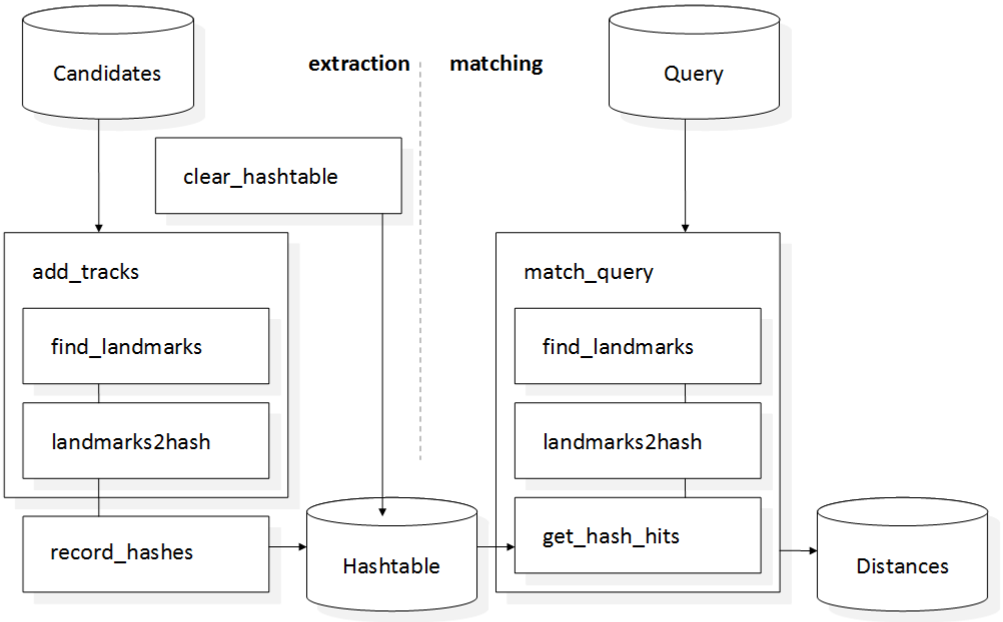


Figure 2.10: Block diagram overview of the landmark fingerprinting system as implemented by Ellis [4]. Mind the separation of extraction and matching stages. Each block represents a Matlab function of which the function should be clear by the name.

3. *Make a uniform selection of spectral peaks ('constellation').*

   (a) Estimate an appropriate threshold thr for peak selection, based on the first 10 peaks.

   (b) Store all peaks of $S(t, f)$ higher than $thr(t, f)$ in a set of (t, f) tuples named pks.

   $$\left.\begin{array}{l} S(t,f) \text{ is a peak} \\ S(t,f) > thr(t,f) \end{array}\right\} (t,f) \rightarrow pks.$$

   (c) Update thr by attenuating it with a decay factor dec and raising it with the convolution of all new pks with a spreading function spr.[9]

   $$thr(t,f) = \max \Big(\; dec \cdot thr(T-1,k), \quad pks(t,f) * spr(f) \; \Big)$$

   (d) Repeat steps (b) to (d) for $t = t + 1$ until all frames are processed.

   (e) Repeat steps (a) to (d) but from the last frame back and considering only (t, f) tuples already in pks. This is referred to as 'pruning'.

Important parameters governing the number of extracted pairs are

[9] This is an approximation, in reality the update is performed every time a new peak is found.

  - The decay factor *dec* (0.998), a function of the wrapping variable *dens* (10).
  - The standard deviation *dev* of the Gaussian spreading function, in bins (30).

4. *Combine nearby points* (t_1, f_1) *and* (t_2, f_2) *into a pair or 'landmark'* L.
   Parameters governing the number of extracted pairs are

   - The pairing horizon in time Δt_{max} (63 frames)
   - The pairing horizon in frequency $\Delta f_{max} > 0$ (31 bins).
   - The maximum number of pairs per peak *ppp* (3).

   $$L = \{t_1, f_1, f_2, \Delta t\}$$

   with $\Delta t = t_1 - t_2$. All f (in bins) and t (in frames) are integers with a fixed higher bound. Due to the horizons, Δt is limited to a lower value than t_2.

5. *Combine* f_1 , f_2 *and* Δt *into a 32-bit hash* h.

   $$h = f_1 \cdot 2^{(m+n)} + \Delta f \cdot 2^n + \Delta t$$

   with $\Delta t = t_1 - t_2$ and m and n the number of bits needed to store Δf and Δt, respectively.

   $$\begin{aligned} m &= \lceil \log_2 \Delta f \rceil \\ n &= \lceil \log_2 \Delta t \rceil \end{aligned}$$

6. *Combine* t_1 *and the song's identifier* C *into a 32-bit unsigned integer* I.

   $$I = C \cdot 2^{14} + t_1$$

   where 14 is taken as t_1's bit size.

7. *Store* I *in the database hash table at index* h.

Table 2.5 shows the names of the Matlab functions that implement the above algorithm steps. Figures 2.10 and 2.11 show the same functions in a block diagram.

Matching

Implementation details for the matching algorithm are given in Algorithm 2.4.

Algorithm 2.4

1. *All the query file's hashes* $\{h\}$ *are extracted as described in Algorithm 2.3.*
   Note that all parameters from Algorithm 2.3 can theoretically be configured independently for extraction and matching. However, the only parameter to which easy access is provided by default is

Steps	Matlab functions
1 2 3 4	`find_landmarks`
5	`landmark2hash`
6 7	`record_hashes`

Table 2.5: Implementation by Ellis [4] of the algorithm steps as described by Wang [3]. The algorithm steps relating to extraction (on the left) are implemented in three Matlab functions (on the right) that can be found on the block diagram in Figure 2.10 and 2.11.

- the peak density parameter *dens* (controlling the decay factor *dec* of the masking threshold)

2. *All database hashes $\{h_d\}$ matching the query's set of hashes $\{h\}$ are retrieved, with their song id's $\{C_d\}$ and timestamps $\{t_{1d}\}$.*

3. *For each song $\{C_d\}$ referenced in $\{h_d\}$ the differences $\{t_{1d} - t_1\}$ are computed.*

4. *If a significant amount of the time differences for song C_d are the same, there is a match.*
   As said earlier, no hypothesis testing is done. There is no decision between match or no match. The matching algorithm just returns the most frequent time offset and the resulting number of matching landmarks for every database song.

$$\begin{cases} \tau_C & = \text{mode}\,\{t_{1d} - t_1\} \\ m_C & = \text{freq}\,\{\tau_C\} \end{cases}$$

   Where *mode* refers to the statistical moment returning the most frequent element in a set and *freq* returns its frequency. The obtained m_C is the number of matching landmarks for database song C, it can be used to compute inter-song distances.

Table 2.6 shows the names of the Matlab functions that implement the above steps of Algorithm 2.4. Figures 2.10 and 2.11 show the same functions in a block diagram.

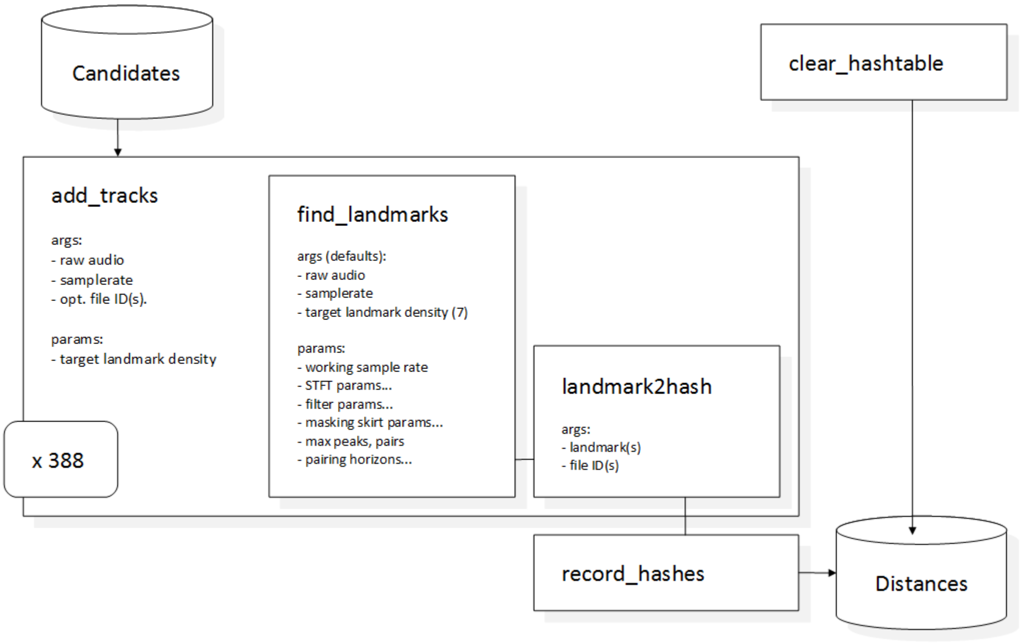


Figure 2.11: Closer look at the extraction stage of the landmark fingerprinting algorithm. Arguments and parameters are indicated for the most important blocks.

Steps	Matlab functions
1	`find_landmarks` `landmark2hash`
2 3	`get_hash_hits`
4	`match_query`

Table 2.6: Implementation by Ellis [4] of the algorithm steps (see Algorithm 2.4) as described by Wang [3]. The algorithm steps relating to matching (on the left) are implemented in four Matlab functions (on the right) that can be found on the block diagram in Figure 2.10 and 2.12.

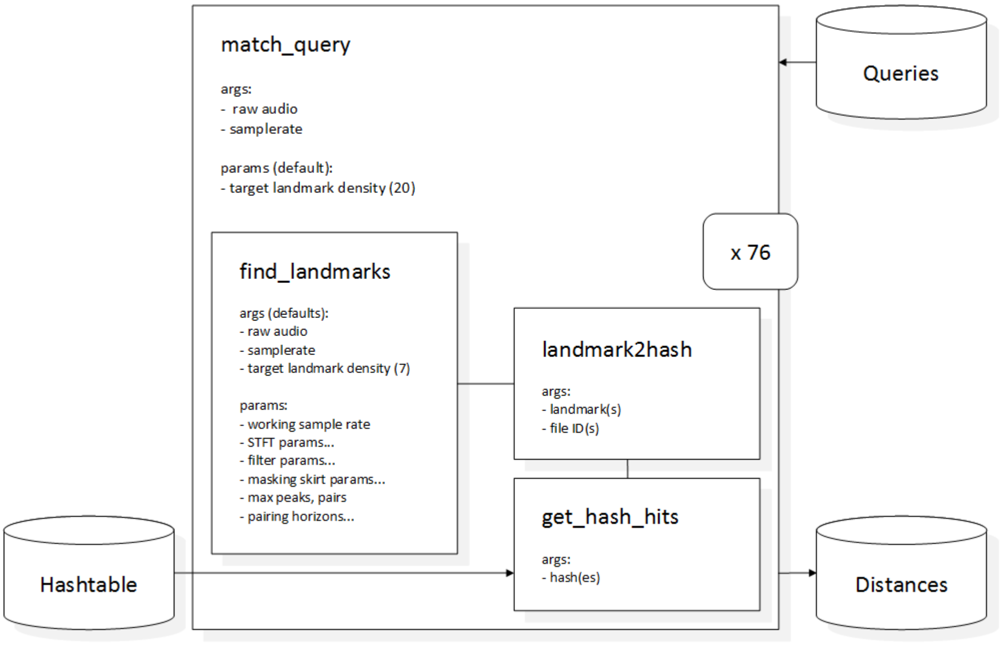


Figure 2.12: Closer look at the matching stage of the algorithm. Note that many of the components are the same as in the extraction stage. The queries are represented as a database for later convenience.

Chapter 3

Evaluation Methodology

This chapter describes the evaluation strategy. Designing an evaluation methodology includes the assembly of a ground-truth database, the selection of a set of evaluation metrics and the implementation of the evaluation scripts that calculate these metrics from retrieval results. Also in this chapter, the evaluation metric's random baselines are computed using randomly generated values instead of actual results. They will be used as a reference in later evaluation.

3.1 Music Collection

The first step towards the development of a sample recognition system is a ground truth database. A good ground truth database is essential for testing and final evaluation. It should be representative and complete. In practice, this means the collection should be large enough to contain at least a few examples of most common types of sampling found. Some annotations should also be made, not only comments as a reference for later, but also quantifiable properties for evaluation. The complete database, including all queries, candidates and samples, can be found in Appendix B.

3.1.1 Structure

Tracks

In this report, the songs that contain one or more samples are referred to as 'queries', and the songs that have been sampled are 'candidates' (see section 3.1.1). All query and candidate tracks in the database are labelled starting with T, the first track being T001 and the last one T199 (not all numbers in between are taken and queries and candidates are mixed). The corresponding audio files are stored in MP3 and WAV format with this label as their file name. All tracks are annotated with:

	Artist	Title	Year	Genre
T034	Pete Rock & C.L. Smooth	Straighten it Out	1992	Hip-hop
T035	Ernie Hines	Our Generation	1972	R&B/Soul

Table 3.1: Example: two tracks as they are represented in the database. Fore more examples, see Appendix B

- Artist
- Track title
- Year of album release
- Genre

Genre can be either Hip-hop, R&B/soul, funk, jazz, world or rock. Table 3.1 illustrates how two tracks are represented in the database. There is a total of 144 query and candidate tracks in the database.

Samples

Each occurrence of a portion of one track in another is a sample. Each sample has a label between S001 and S137 and references two track plus a set of annotated time stamps.

- Sampled track (C)
- Track in which the fragment appears as a sample (Q)
- Time at which the fragment occurs in the candidate (TC) [1]
- Time at which the fragment occurs in the query the first time (TQ)
- Number of times the sample appears in the query (N)
- Optional comments (e.g. 'maybe interpolated', 'short', 'long'...)

If a query samples two fragments of the same candidate, two samples will refer to the same C and Q. Of the 144 referenced tracks, 76 are query and 68 candidate files. Table 3.2 gives an example of a sample as it is represented in the database. There is a total of 104 samples in the database.

[1] The resolution of time annotations was 1 second, not only because this was the accuracy of available annotations on specialized websites and of the playback software.

	C	Q	TC	TQ	N	Comments
S019	T035	T034	0:40	0:10	48	vocals

Table 3.2: Example of a sample as it is represented in the database. Fore more examples, see Appendix B

Noise files

To make the retrieval task more challenging, 320 noise files were added to the database and labeled N001 to N320. These files are songs that are similar to the candidate songs in genre, length and audio quality (some of them even contain sampled breaks), but have not been sampled by any of the queries. A match with any of them is therefore a false positive. With the noise files, there is a total of 464 tracks in the database.

3.1.2 Content

Scope of the Collection

The outlines for the assembly of the ground truth database are listed in the form of three important restrictions. The most important reason to set some limitations to the collection content is that a database cannot be representative to a range of concepts that is too wide. Also considerably important is that the collection is built manually. Time and resources are required to explore and understand the music and to collect and annotate a large amount of files, and selectivity makes this a feasible task.

1. Only hip hop songs are considered as queries. As said in section 1.2, many genres in popular and electronic music make use of samples, but this work restricts to just one. Hip hop is chosen because the genre and the practice of sampling are connected in their very origins. As always in genre labeling, some confusion is possible, yet especially so for songs that sample from other genres. Here, only songs that show clear hip hop characteristics, i.e. rap vocals and samples, are considered.

2. Only direct samples are considered. Direct sampling involves audio from the original sampled record. ‘Interpolations’ (see 1.2) or samples that are probably interpolated, are not considered. It is often very hard to tell if a sample is a direct sample or not. Expert sources do not exist except for the artist himself. Therefore, some of the tracks in the database have been annotated with a ‘maybe interpolated’ label, for possible later reference.

3. Only samples on the order of one second or longer were considered, as annotations were done manually on a time scale limited to seconds. Short samples (as short as one second) were only included if they were repeated several times throughout the query.

The ground truth was composed using valuable information from specialized internet sites. Especially the websites Whosampled[2] and Hip hop is Read[3] proved particularly useful. The former provides the possibility to listen to songs and provides user-submitted information about the time at which the sample occurs in both query and candidate, the latter contains a lot of background information.

Representativeness

In the construction of the ground truth, special attention was given to representativeness. Some distinctions were made to identify types of samples, so an eye could be kept on the share of each of those categories in the whole of the database:

Drum samples versus tonal samples. During the compilation of this collection, it often proved difficult to obtain a high quality digital version of candidate files. A good source is the Extended Player collection 'Ultimate Breaks and Beats' mentioned in section 1.2, but these records mainly contain drum breaks. To keep the database representative, some effort was put into including more samples than just those found on this kind of compilations. This was also important not to bias the database towards older examples, dating from the days in which the series were particularly popular.

Long samples versus short samples. Samples are not always easily recognizable. There is an entire on-line community of people devoted to the identification and documentation of samples in all kinds of music. One can assume that long samples are generally easier to recognize than short ones, or at least easier to distinguish from similar samples. Therefore, the collection made for the ground truth could be biased in favour of longer, perhaps more interesting samples. In attempt to prevent this, some very short samples were added (e.g. entries S046 and S056 in Appendix B). The longest sample in the database is 26 seconds long and repeated 5 times, the second longest sample is less than 12 seconds long.

Isolated samples versus background samples. Isolated samples, i.e. samples that appear at least once with little or no extra musical layers on top of it, are easier to recognize as well. To keep the ground truth representative, some of the included samples are obscured by a rather thick layer of other samples or instruments (e.g. S084).

Other aspects such as amount of time-stretching, re-pitching and effects are not taken into account. It is assumed that not giving any special attention would result in a random but representative distribution of properties. Generally, samples and originals were included if they could be recognized as the same by a human listener.

[2] `www.whosampled.com`

[3] `www.hiphopisread.com`

3.2 Evaluation metrics

General evaluation procedures make a distinction between queries and candidate files. Every query (hip-hop song) is fed to the system . The system then returns matrices containing the variables computed for every query vs. candidate/noise file pair. Three evaluation metrics were chosen to quantify the system's performance.

The Mean Average Precision (MAP) was chosen as the main evaluation measure. The MAP is a common evaluation measure for information retrieval tasks [40]. It works on the distance matrix, i.e. the matrix with computed distances between all query and candidate/noise files, and is computed as follows.

1. For every query Q, sort the candidate/noise files C from lower to higher distance.
2. At every correctly retrieved document C_m in Q, the precision so far is calculated.
3. Average the obtained precisions over all C_m.
4. Compute the mean of these average precisions over all Q to obtain the MAP.

Two more evaluation measures were conceived to measure the performance of the system in localizing samples within a song. Ultimately, the metrics were not used, but annotations were made and added to the logged samples. This information could be very valuable in future work on (the identification of) hip hop samples.

The mean errors in TC and TQ. For every sample, the times TC and TQ at which the sample occurs in candidate and query is estimated to obtain the errors ETC and ETQ. The mean of these errors is computed over all correctly retrieved elements and all queries.

The mean error in N. For every sample, the number of occurrences N is counted to obtain the error EN. The mean of this EN is then computed over all correctly retrieved elements and all queries.

An evaluation function is implemented in Python to calculate these measures from a distance matrix (for the MAP), a TC and TQ matrix, and a matrix with estimates for N, plus the ground truth. The dimensions of these matrices are the number of queries by the number of candidates including noise (76 × 388).

	MAP	ETC	ETQ	EN
Baseline	0.017	103.40	98.51	3.57
Standard deviation	0.007	6.90	6.62	0.79
Minimum value	0.008	87.77	80.57	2.16
Maximum value	0.043	119.66	115.66	5.50

Table 3.3: Random baselines for the proposed evaluation measures and the ground truth database. Results summarized from 100 iterations.

3.3 Random baselines

The performance of an information retrieval system should always be seen in the context of the used data. Precision and recall depend on the size of the candidate collection and if this set contains noise or not. In an ideal evaluation, the set of candidates is infinite. A random selection of matches will therefore yield a precision and recall of both zero. In a realistic experiment however, this random baseline can be significant. This is why noise files are added to the candidate set and the random baselines are calculated as a reference for later results.

The random baselines corresponding to these evaluation measures were computed by generating 100 random matrices for each of the variables involved.

- Distances for the MAP were uniform between 0 and 1
- T1 and T2 were uniform between 0 and the length of the track
- N (because of its complex distribution) were random permutations of the true occurrence counts

A statistical summary of the 100 iterations is given in the table 3.3.

Chapter 4

Optimisation of a State-of-the-art System

Chapter 2 gives an overview of the most relevant work done in what could be called 'music identification'. The ground truth dataset's random baselines have been summarized in Table 3.3. However, apart from the random baselines, there needs to be a state-of-the-art reference to which results are ultimately compared in the evaluation. For this purpose, the most promising of the reviewed systems was put to the test and optimised in a first experiment involving all queries and the complete database of candidates and noise files to obtain the state-of-the-art performance. The observed parameter in this optimisation is the MAP.

4.1 Optimisation of the Landmark-based Audio Fingerprinting System

The chosen system is the landmark based audio search algorithm by Avery Wang [3] described in section 2.4.4. The reasons why this implementation has been chosen come down to its advantages as they were described in Table 2.4.

- High robustness to several types of noise and distortion.
- Ability to identify music from only a very short segment.
- Does not explicitly require tonal content.

4.1.1 Methodology

The implementation by Ellis [4] described in section 2.4.5 has been used. Extensive experiments were carried out to optimise parameters of the system. However, some adaptations had to be made.

Adaptations

First, as the input of the evaluation script requires a distance matrix, a distance function had to be defined. The most straightforward option is chosen: the distance between two files is inversely proportional to the absolute number of matching landmarks (hence d_a). For any query and landmark pair (Q,C):

$$d_a = \frac{1}{m_C + 1}$$

where m_C is the number of matching landmarks for candidate C.

It has proved helpful to define an alternative distance, normalising the number of matching landmarks by the total number of landmarks extracted (hence d_n). It is determined by the fraction of extracted landmarks that match.

$$d_n = 1 - \frac{m_C}{l_Q}$$

where l_Q is the number of landmarks extracted from the (sliced) query Q. On average, the choice of distance measure did not significantly influence the results of experiments. Rather, the distance serves as a useful second opinion, so both distances will normally be reported.

Second, wrapper functions had to be written to feed the audio to the algorithm. The algorithm has been designed to handle candidates of lengths on the order of minutes, and queries of lengths on the order of seconds. The wrapper functions ideally reflect this difference. Yet this is not a drawback, it actually suits the problem. Sample identification does not aim to obtain the global similarity between two files, it aims at understanding which particular segments are alike.

A first wrapper function written in Matlab extracts all the candidate and noise files to the database. A second wrapper slices all query audio to chunks with length N_W, computed at every hop of length H_W, and feeds those to the system one by one.

Choice of Parameters to Optimize

The system in [4] is very flexible, but in optimisation this causes some problems. As experiments take up to several hours with the available computational resources, there is no time to study the effect of all parameters in all possible configurations. To make a selection, the adapted system's parameters can roughly be divided into three groups, as shown in Table 4.1: parameters governing the properties of the landmarks, parameters regulating the number of landmarks and parameters of the query windowing. Please refer to section 2.4.5 for more details about the role of every parameter.

N_W H_W	Parameters governing the slicing of the query
SR N H Δt_{max} Δf_{max}	Parameters governing the properties of the landmarks
ppp $dens$ dev	Parameters governing the number of landmarks

Table 4.1: Parameters of the (adapted) implementation of the landmark-based audio search system by Wang (see section 2.4.5 for details). They can roughly be divided into three categories.

1. The parameters N_W and H_W governing the slicing of the query have to be optimised without any doubt. There are no defaults and the effect of changing them is unknown.

2. Changing the parameters that make up the properties of the landmarks generally has an effect on the hashing procedure. Actions such as increasing the bandwidth (higher SR) or the frequency resolution (higher N) will lead to a bigger fingerprint size. The choice by Ellis to work with 22-bit landmarks is not only implemented in a rather inflexible way, it is also a reasonable compromise in itself. Adding more bits to the hash would allow for less landmarks to be stored in the same bucket when the total hashtable size is limited to that of the largest matrix storeable in the RAM of an average computer running Matlab with default memory settings. For this reason, the parameter SR, N and H are left unchanged in the optimisation experiment.

3. Changing parameters that control the number of landmarks is a valid focus for optimisation. The parameters $dens$, ppp and dev will be included in the set of parameters to be optimised. It is an interesting property of these parameters that the number of extracted landmarks can be greater for the query than for the candidate. This could be done to save on fingerprint size, but also reflects the reality of many types of fingerprinting tasks: the query contains the most information, and only a part of it relates to its matching candidate. This allows to emphasize on the matching process and keep the candidate database the same, so that more experiments can be done.

Starting with the default settings, the chosen parameters (N_W and H_W, $dens$, ppp, dev) are now changed incrementally and one by one, until a maximum MAP is found.

N_W	H_W	*ppp*	*dens*	*dev*	MAP_n
2	1	3	10	30	0.038
4	1	3	10	30	0.055
8	1	3	10	30	0.060
12	1	3	10	30	0.089
16	1	3	10	30	0.079

Table 4.2: Results from the optimisation of the query chunk size N_W. A sparse set of lengths is chosen as each experiment with $H_W = 1$ takes several hours.

4.1.2 Results

Optimisation of (N_W, H_W).

The first step is the optimisation of N_W. Recall that every query file is fed to the matching script as a series of segments of length N_W (in seconds), with an overlap determined by the hop size H_W (in seconds). The time resolution of annotations is limited to one second, therefore a maximum overlap was chosen by setting $H_W = 1$. An optimal window size was then found by varying N_W and running a complete experiment with all queries and a full database at every step. Two remarks must be made.

- Experiments with $H_W = 1$ took up to 10 hours, so only few runs could be done within the available time.
- Only one distance function d_n was initially in use: only one MAP is computed for every complete run.

An optimal window size of 12 seconds is found. Results are shown in Table 4.2.

In the next series of experiments, H_W was increased to 6 (or 50% overlap) in order to perform experiments faster. This decision is based on two assumptions.

1. The effect of changing the query start time is independent of parameter choices for *ppp*, *dens* and *dev*.
2. An overlap of 50% does not leave any information unused. Some samples may no longer be represented entirely in one query, but this goes only for samples longer than 6 seconds.

Optimisation of *ppp*

The next parameter that has been optimised is the *ppp* of the query fingerprint. A series of experiments is performed with default extraction parameters but a varying *ppp* at the

N_W	H_W	*ppp*	*dens*	*dev*	MAP_n	MAP_a
12	6	1	10	30	0.093	0.122
12	6	2	10	30	0.099	0.108
12	6	3	10	30	0.116	0.114
12	6	4	10	30	0.111	0.114
12	6	5	10	30	0.115	0.114
12	6	7	10	30	0.116	0.114
12	6	10	10	30	0.110	0.117
12	6	15	10	30	0.110	0.116

Table 4.3: Results of the optimisation of the target number of pairs per peak *ppp* for the query fingerprint. The candidate extraction parameters were kept default.

N_W	H_W	*ppp*	*dens*	*dev*	MAP_n	MAP_a
12	6	10	16	30	0.103	0.098
12	6	10	20	30	0.117	0.110
12	6	10	22	30	0.121	0.137
12	6	10	24	30	0.121	0.141
12	6	10	25	30	0.127	0.128
12	6	10	28	30	0.127	0.122
12	6	10	32	30	0.128	0.111
12	6	10	36	30	0.133	0.118
12	6	10	40	30	0.124	0.111
12	6	10	44	30	0.125	0.112

Table 4.4: Results from the optimisation of the target landmark density *dens* of the query fingerprint. The candidate extraction parameters were kept default.

matching stage. Recall that *ppp* is the number of pairs that can be formed per peak. It is set by default to 3. After testing, an optimal target *ppp* is found to be 10. Results are given in Table 4.3.

Optimisation of *dens*

The third parameter that is optimised is *dens*. It controls the masking decay factor *dec*. In the peak selection process, this main parameter governs how far the applied masking extends over time. The resulting density parameter *dens* is optimised in a series of experiments involving the same default extraction parameters, but using $ppp = 10$ and a varying *dens* for matching. The results are displayed in Table 4.4: an optimum is found in $dens = 36$.

N_W	H_W	ppp	$dens$	dev	MAP_n	MAP_a
12	6	10	36	10	0.111	0.113
12	6	10	36	15	0.123	0.113
12	6	10	36	20	0.122	0.129
12	6	10	36	25	0.109	0.123
12	6	10	36	30	0.133	0.118
12	6	10	36	35	0.114	0.105

Table 4.5: Results from the optimisation of the query fingerprint's *dev* parameter, controlling the extension of masking in the frequency dimension. The experiments show that the default value std = 30 is also optimal.

N_W	H_W	ppp	$dens$	dev	MAP_n	MAP_a	Random Baseline
12	1	10	36	30	0.147	0.128	0.017 (0.007)

Table 4.6: State-of-the-art baseline with parameters of the optimised landmark fingerprinting system. The random baseline (mean and std) are provided for reference.

Optimisation of *dev*

As a fourth and last step in the optimisation, *dev* is optimised. It is the standard deviation of the Gaussian spreading function with which the peaks are convolved to obtain the updated masking threshold. It controls the extension of masking in the frequency dimension and defaults to 30 bins. In the last set of experiments *dev* is varied for the extraction of the query fingerprint, while candidate extraction parameters are kept default. The results in Table 2.4 show that the default value 30 is also optimal and that the influence of this parameter is minor.

The outcome of all described experiments are summarized in Table 4.6 listing the optimised parameters and the resulting state-of-the-art baseline. The final baseline has been obtained by repeating the most promising of above experiments with $H_W = 1$.

4.1.3 Discussion

A few remarks situating these results:

- The presented experiments cannot be regarded as a *complete* optimisation of the system. First, the parameters that have not been included in the optimisation could have an effect on performance, and second, the optimised parameters were only optimised for the matching stage for reasons made clear in section 4.1.1.
- Concluding that the found optimal values will always hold as an optimum in different contexts would also be too bold a claim. Parameters of a complex system can generally cannot be expected to behave in a completely independent manner.

	Q	C	Instruments	Tonal	Transposed	Comments
S001	T001	T002	bass, drums	yes	no	
S005	T008	T007	several	yes	no	
S015	T026	T027	several	yes	no	
S038	T063	T062	several	yes	no	very long
S052	T086	T088	drums	no	no	short
S061	T099	T098	drums	no	no	never isolated
S067	T109	T108	drums	yes	no	very soft synth
S079	T146	T145	drums	yes	no	very soft piano
S101	T176	T177	bass, drums	yes	no	
S103	T179	T180	several	yes	no	
S107	T184	T083	drums	no	no	never isolated
S112	T187	T088	drums	yes	no	toms present

Table 4.7: Overview of the samples that were correctly retreived top 1 in the optimised state-of-the-art system, and some of their properties.

However, a consistent and thorough effort to maximize the performance has been done. This being said, it is interesting to point out that the obtained optimum for N_W makes sense as a time scale on which to observe samples. As said in section 3.1, all but one of the samples in the ground truth database N_W are shorter than 12 seconds.

Overall, the obtained state-of-the-art baseline is clearly far from the random baseline for the MAP, several times greater than the maximum over 100 random iterations. This shows that the basic principle of spectral peak-based fingerprinting effectively works in some cases. To see in which cases it worked and for which queries it didn't, the next paragraph analyses the optimised performance in terms of recognised and unrecognised samples.

Performance analysis

It is useful to take a closer look at the results obtained with the optimised system. The main question to ask is: which kind of samples are retrieved and which are not? The retrieval system computes only similarities, it does not have a threshold to distinguish between retrieved and not retrieved. For this reason, a script was written in Matlab to extract the query and candidate IDs Q and C for all correct top 1 retrievals. The following was observed:

- 12 out of the 76 queries retrieve a relevant document as a best match. The 12 samples involved are listed in Table 4.7, with some useful extra information annotated after listening.
- All of these samples appear at original pitch and tempo. Time-stretched or transposed audio is not recognised.

- Both drum and tonal samples have been recognised. This suggests that, even though being spectral peak-based, the system is able to deal with queries with a generally flat spectrum such as drum recordings. However, 3 of the 6 samples annotated with 'drums' actually do contain some pitch, either from the very subtle presence of another instrument, or from toms (see Table 4.7). Also, 4 of them appear isolated. Hence, no conclusion can be made about the ability of the system to recognise pure percussive sounds in a tonal context.
- Still only a small minority of samples is identified. Apart from re-pitch detection and problems identifying drum loops, some other issues could still be at play: most detected samples are rather long and not buried too deep into the mix. Possibly sample length and amount of noise contribute to the challenge.

More could be learned from a look at those samples that are not retrieved, but in this stage of the work, these are very numerous. Also, apart from those retrieved as a best match, only two more relevant documents were retrieved as top 5. Listening to the confused files could not clarify why these matches were 'almost found', but not entirely.

Conclusions

The three main conclusions that can be drawn from this optimisation experiment are:

1. An optimised state-of-the-art fingerprinting system has been found able to, in a small number of specific cases, recognise real-life examples of sampling. Performance significantly above random has been achieved.
2. Transposed and/or time stretched audio are not recognised by the fingerprinting system.
3. Drum samples have been identified, but more experiments are needed to make conclusions on the identification of percussive sounds.

Chapter 5

Resolution Experiments

The following experiments propose changes to the fingerprinting strategy in an attempt to increase the state-of-the-art performance. The new strategies are tested and the results are discussed. Along the way, the challenges will become more clear with every experiment. For now, the experiments have been set-up to deal with two major challenges:

1. Transposed and time-stretched versions of the same sounds need to be identifiable as the same sound.

2. Percussive sounds must be fingerprintable.

The first of the following sections reports on the importance and behaviour of some of the parameters that were not considered in the optimisation. In the second section, a more drastic change is made to the fingerprinting strategy by proposing the use of a constant Q transform to obtain the spectral representation. Later experiments investigate some possible ways to deal with transposed and time-stretched samples.

5.1 Frequency Resolution and Sample Rate

To further assess if the system is able to recognise pure percussive samples, new changes have to be made to the algorithm explained in section 2.4.5 and optimised in Chapter 4. A promising set of parameters to turn to is SR, N and H. These parameters have previously been left out of the optimisation because changing them would affect the hash. The parameters SR and N can for this reason not be increased. However, they can be decreased, or varied together.

5.1.1 Motivation

For the sake of fingerprint transparency, i.e. the capability of the system to distinguish the presence of several sounds at the same time, lowering the frequency resolution intuitively seems counterproductive. However, a look at the drums samples inside some of the queries has shown that:

- Many of the loops consist for the most part of distinct strokes with a clear onset. These strokes might mask other sounds present during the short time at which their energy peaks.

In the spectral domain, it has then been observed that:

- Drum sounds generally have a noisy spectrum. This means that there are only few well-defined spectral peaks and few or none are regularly spaced. Compared to a tonal sound, a drum sound is stochastic and is therefore less defined by its spectral details, yet all the more by its spectral envelope.
- From inspecting the spectrum of a number of drum strokes, it is observed that the defining spectral elements of bass and snare drum sounds are in the lower frequency range. Indeed, the frequency regions of highest energy for a bass and snare drum should be 0-150 Hz and 100-500 Hz, respectively [41]. Spectrums such as the one in Figure 5.1 reflect this.

The intention here is obviously not to describe or discriminate between bass drum and snare sounds. Nevertheless, knowledge of the spectrum can be useful. For example, in perceptual coding, several technologies make use of decomposition of the signal into sinusoids and noise [42]. Noise can then be coded by its spectral envelope (for examples as a series of ERB or Bark band energies) to save bits, as in [43]. Under this type of coding, any spectral details on finer scale than the spectral envelope will be lost.

This is the motivation for the first of the following performance studies, in which N and H are decreased, thus lowering the frequency resolution, but increasing the time resolution. As an example, computing the spectrum with a window of $N = 32$ ms at $SR = 8000$ Hz yields a positive magnitude spectrum of 64 bins. Corresponding to a bandwidth of around 64Hz per bin, this value approaches the bark band width for low frequencies, classically approximated as 100Hz for central frequencies below 500 Hz [44].

The observations also inspired for a second series of experiments, in which the sampling rate SR is lowered to extract more landmarks from those spectral regions in which bass and snare drums are classically discriminated. No hypothesis is formulated for any of these experiments, the simple objective is to study the effect of the involved parameters on the overall system performance.

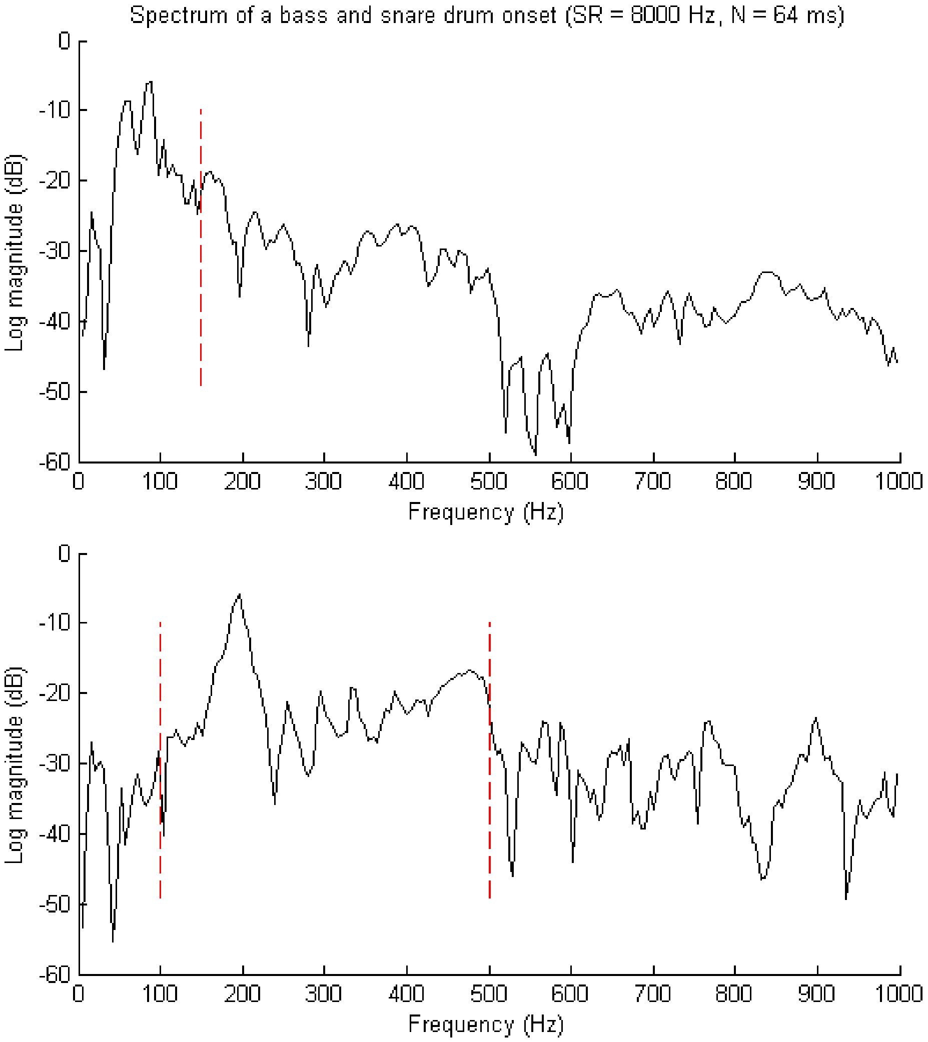


Figure 5.1: Spectrum of a bass and snare drum onset extracted from track T085 (Isaac Hayes - The Breakthrough) (SR = 8000 Hz, N = 64 ms). Frequencies up to 1000 Hz are shown. The dashes indicate the 150 Hz line and the 100 and 500 Hz lines, respectively.

N_W	H_W	ppp	$dens$	dev	SR	$N(ms)$	$H(ms)$	MAP_n	MAP_a
12	6	10	36	30	8000	64	32	0.133	0.118
12	6	10	36	30	8000	32	16	0.105	0.100
12	6	10	18	15	8000	32	16	0.122	0.111
12	6	10	36	30	8000	16	8	0.089	0.087
12	6	10	18	15	8000	16	8	0.132	0.116
12	6	10	9	7.5	8000	16	8	0.123	0.115

Table 5.1: Results of experiments varying the FFT parameters N and H. The previously optimised parameters were kept optimal. In some experiments, the masking parameters *dens* and *dev* are adapted to reflect the changes in frequency and time resolution, but keeping the total density of landmarks the approximately same.

5.1.2 Results

Lower frequency resolution

Several experiments have been carried out to study the effect of decreasing N and H. With a constant overlap of 50%, N was lowered to 1/2 and 1/4 of the original window size. This way, the frequency resolution is halved, but the time resolution is doubled. The results are given in Table 5.1. The changes have a negative effect on performance.

One could argue that the parameters governing the number of landmarks should be adapted to the new spectrum size. Note that even though the target peak density may appear to doubled because of the smaller hop size H, it is not, as only half the number of bins remain. However, the masking skirt has been optimised to have a deviation *dev* of 30 bins in either direction from each found peak. This way it may span the whole spectrum of only 64 bins. For this reason, extra experiments were performed with adapted *dev* and *dens* parameters, preserving the amount of masking but reshaping the masking skirt. They appear as the the last two entries in the table.

The performance measures give an indication of how well samples are recognised, but do not reflect which samples are recognised best. A spreadsheet was created to keep track of all samples retrieved as top 1, in all the experiments performed from this chapter on. The following remarks summarize the most important findings on the recognition coverage in this section's experiments.

Experiments lowering the frequency resolution:

New samples retrieved by the best performing experiment:	3
New samples retrieved in the whole experiments series:	6
New transposed samples retrieved:	2
New drum samples retrieved:	0

N_W	H_W	*ppp*	*dens*	*dev*	SR	$N(ms)$	$H(ms)$	MAP_n	MAP_a
12	6	10	36	30	8000	64	32	0.133	0.118
12	6	10	36	30	4000	128	64	0.203	0.190
12	6	10	36	30	4000	64	32	0.193	0.193
12	6	10	36	30	2000	256	128	0.133	0.144
12	6	10	36	30	2000	128	64	0.218	0.228
12	6	10	36	30	2000	64	32	0.162	0.176
12	6	10	36	30	1000	256	128	0.131	0.131
12	6	10	36	30	1000	128	64	0.201	0.218
12	6	10	36	30	1000	64	32	0.151	0.176

Table 5.2: Results of experiments varying the sample rate SR. Where possible, N and H were varied to explore the new trade-off options between frequency and time resolution.

Downsampling

Experiments have then been carried out to study the effect of decreasing the sample rate SR. It was lowered to 1/2 and 1/4 of the original window size. The results are given in Table 5.2. Lowering the sample rate comes with a loss of information, resulting in a number of trade-off options between frequency and time resolution. Some of these (N, H) options were tested as well. The changes have a positive effect on performance.

Experiments lowering the sample rate:

New samples retrieved by the best performing experiment:	7
New samples retrieved in the whole experiments series:	14
New transposed samples retrieved:	0
New drum samples retrieved:	5

5.1.3 Discussion

A modest attempt is made to study the possibility of moving towards a landmark-based fingerprinting system with reduced frequency resolution but increased time resolution, in order to fingerprint percussive sounds. As it is implemented here, the strategy does not seem promising. Performance was lower, and no new drum samples were found. The recognition of a small number of new samples has been observed but their number is too low to conclude that their identification can be attributed to the novel strategy.

This being said, it is worth noting that two transposed samples were correctly retrieved. Both samples have been transposed one semitone down and appear to be a corresponding 6% slower than their original, suggesting they have been repitched. A possible 'explanation' is that the matching landmarks originate in the lowest frequency range. Sinusoids in the lowest 10 or so bins may be mapped to the same bin under a transposition of only 6%, and the lower the frequency resolution, the higher the central frequency of this 10th bin.

An attempt has also been made to investigate the effect of lowering the sample rate. The strategy is more promising than lowering the frequency resolution. The best performing configuration of parameters involves a sample rate of $SR = 2000$ Hz, and FFT parameters $N = 128$ and $H = 64$ ms. It achieves a mean average precision of 0.218, significantly better than the best-so-far $MAP = 0.133$ (both using distance d_n).

Note that downsampling discards information. The optimised FFT parameters reflect this: they correspond to a comprimise between frequency loss and time resolution loss: the 128 point positive spectrum is smaller than the one used in Chapter 4, and the frame rate (16 Hz) is halved as well. Experiments in which either of the resolutions were conserved performed worse.

A thorough explanation for the enhanced performance would require many more tests. The loss of information has a large variety of implications. It affects, amongst others, the amount of information (or entropy) in the landmarks, the resolutions in terms of bin width and frame rate, and the masking procedure in the peak selection. Trying to explain the performance increase in terms of all these aspects is not in the scope of the experiments. Instead, this section is round up concluding that:

1. Fingerprinting audio at a lower sample rate has been found to increase the performance of the fingerprinting system significantly.

2. Time and frequency resolution is lost but a satisfying trade-off in this loss has been found.

5.2 Constant Q Landmarks

5.2.1 Motivation

A completely different, but common way to analyse spectral information is to study it in terms of logarithmic frequency. A logarithmic division of the frequency axis is used in the constant Q transform (see section 2.1.2) and equal-tempered PCP's (see for example [33] in section 2.3.1), or as a cepstrum representation in LFCC's (same example). Haitsma [30], as said in section 2.4.3, represents the signal as energies in a series of logarithmically spaced bands.

A logarithmic division of the frequency axis has proved useful because it reflects the behaviour of the HAS. The general perception of frequency operates on approximately logarithmically spaced bands [30]. On the finer scale, it suits the geometric spacing of the frequencies of tones in Western scales [29]. The constant Q transform has also proved useful outside of fingerprinting, for example in detection of musical key [45].

In the context of this research, the constant Q transform instead of the STFT serves in the first place as another reorganisation of the frequency resolution to investigate. The approach is tested in the following set of experiments.

N_W	H_W	f_{min}	bpo	SR	$N(ms)$	$H(ms)$	MAP_n	MAP_a
12	6	32	32	8000	64	32	0.172	0.164
12	6	32	32	8000	128	64	0.211	0.170
12	6	32	32	8000	256	128	0.175	0.163
12	6	32	32	8000	128	64	0.211	0.170
12	6	32	32	4000	128	64	0.213	0.195
12	6	32	32	2000	128	64	0.144	0.151
12	6	32	24	8000	128	64	0.197	0.182
12	6	32	32	8000	128	64	0.211	0.170
12	6	100	48	8000	128	64	0.150	0.151

Table 5.3: Results of experiments using a constant Q transform to obtain the spectrum. Three different FFT sizes, three different samplerates and three different resolutions *bpo* have been tested.

5.2.2 Methodology

The constant Q algorithm used was implemented by Ellis[1] [32]. It computes a short term constant Q transform with the parameters listed in section 2.1.2, returning a matrix with columns of length K, where K is the number of geometrically spaced frequency bins. A good resolution (almost 3 bins per semitone) could be combined with a broad frequency range (almost full) by initially setting $f_{min} = 32$ and $bpo = 32$ and restoring $SR = 8000$.

5.2.3 Results

In a first series of experiments the effect of using this constant Q transform was tested with three different FFT sizes N and a constant overlap, analog to the experiments in section 5.1. This was done to find a compromise between loss of time resolution and blurring of the lowest frequencies. The latter is significant if the default $N = 64$ ms is used. In a second series of experiments similar to the one in section 5.1.2, the sample rate was then lowered to see if a similar increase in performance could be observed. In a last series, the experiment is rerun with 2 ($bpo = 24$) and with 4 ($bpo = 48$) bins per semitone. The experiments are summarised in Table 5.3.

5.2.4 Discussion

The results are promising. A good MAP of 0.21 is obtained for $SR = 4000$ Hz and 8000 Hz. Though close, it is not better than the best MAP so far (regular landmarks at $SR = 2000$; $MAP = 0.22$). In terms of retrieval, most of the samples are the same. All new correct retrievals are untransposed samples and the large majority is tonal. Again, the experiment is not a complete optimisation. Some parameters were not tested. Yet

[1] http://www.ee.columbia.edu/ dpwe/resources/matlab/sgram/

a fair effort has been done to identify the most important ones and briefly study their influence.

Chapter 6

Fingerprinting Repitched Audio

No strategies to recognise transposed and time-stretched samples have yet been explored. In this chapter, two possible approaches are presented. However, they are limited to the identification of samples that were time-stretched and transposed by changing its playback speed, i.e. their pitch and tempo are changed by the same factor. This form of transposition will be referred to as ‘repitching’. The restriction is based on the historical development (see section 1.2.2) of samplers and is partly confirmed by observations made in the ground truth database. A majority of samples in the database seems to show that, if the pitch is changed, tempo is changed accordingly. However, this is hard to verify exactly, especially for samples without a clear pitch.

6.1 Repitch-free Landmarks

6.1.1 Methodology

The first strategy consists of using the landmark-based system, but proposes a significant change to the hashing and matching steps. The observation from which it starts is that, in an ideal time-frequency space, time values and frequency values are scaled inversely under a repitch transformation. If time values increase (the sample expands), frequencies and frequency differences go down. Hence, any product of one time and one frequency value will remain the same.

Landmarks as Previously Defined

Recall that a landmark is defined as

$$L = \{t_1,\, f_1,\, f_2,\, \Delta t\}.$$

Its hash is then computed as

$$h = f_1 \cdot 2^{(m+n)} + \Delta f \cdot 2^n + \Delta t$$

with $\Delta t = t_1 - t_2$ and m and n the number of bits needed to store Δf and Δt, respectively. The resulting number is used to point to the location where $\{I_C\}$ is stored, a set of integers containing t_1 and the numeric song identifier s for all songs sharing a landmark with this hash.

Recall also that, as explained in section 2.4.5, matching a query landmarks against the database consists of finding the most frequent time offset τ in the collection of database landmarks with the same hash.

$$\tau_C = \text{mode}\,\{t_{1C} - t_1\}$$

where t_1 is extracted from the query and t_{1C} is retrieved from the database.

Repitch-free landmarks

The above considerations on the invariance of time and frequency products suggest a new type of hash that could be robust to repitching:

$$h = (f_1 \cdot \Delta t) \cdot 2^n + (\Delta f \cdot \Delta t) \tag{6.1}$$

in which n is now the number of bits needed to store the last product. Note how the same three landmark properties $(f_1, \Delta f, \Delta t)$ are used. The time value has to be used twice for complete invariance. Clearly, the matching stage will have to adapt as well, τ is not invariant to re-pitching. Analogue to the redefinition of the hash, τ_C can be redefined as

$$\tau_C = \text{mode}\,\{t_{1C} \cdot \frac{\Delta t}{\Delta t_C} - t_1\}. \tag{6.2}$$

where again Δt is extracted from the query and Δt_{1C} is retrieved from the database. This formula is derived in Appendix A. A new type of landmark can now be defined to contain both products needed for the hash, and the values needed to compute τ. Note that only the last two are the invariant elements.

$$M = \{t_1,\, \Delta f,\, (f_1 \cdot \Delta t),\, (\Delta f \cdot \Delta t)\}. \tag{6.3}$$

Hash Size and Entropy

An efficiency issue arises from this definition. The proposed hash will be several times greater than the original, while its entropy goes down. Indeed, the associativity of the product will map unrelated (f, t) pairs to the same $(f \cdot t)$ value. When the entropy in a fingerprint is too low, there will be a risk of retrieving false positives. Extracting more landmarks per song could be a way to maintain the fingerprint entropy even though the hash entropy goes down, but either way the result is a very inefficient representation. For example, without any adaptations and with the default settings for f and t, the hashtable would become 16 times bigger, easily exceeding the size of the largest matrix storeable in the RAM of an average computer.

Information science provides ways to code messages in an economic manner, where coding efficiency is expressed with respect to the total information contained in the message. This value, expressed in bits by the Shannon entropy, is the theoretical lower limit for the message length (also in bits). Regarding the hashing issue, this means it should be feasible to reduce the hash size to a value on the order of the Shannon entropy. As an example, $f_1, \Delta f$ and Δt could be stored separately, just like it is done in the landmark system. Indeed, for uniformly distributed and independent random integers this will minimize the length of h to its limit.

However, in the proposed strategy, the products are crucial. One straightforward way to reduce the hash size is to map the products to an new alphabet (of integers) that represents only products that can actually occur. This excludes, for example, all prime numbers higher than the highest possible value of f and t. It is clear that this mapping cannot be done analytically, the alphabet has to be constructed by computing all possible products. Though once this is done, it is observed that for realistic ranges of f and t the mapping allows for a reduction of the hash by at least one bit in each product, or 75% for the entire hash.

Implementation

The proposed changes to the landmark-based fingerprinting system are implemented. The proposed strategy requires a linear spacing of the frequencies, so it starts from an STFT-based spectrogram instead of the constant Q-based one used in the last section. The changes that had to be made to the system are illustrated in the block diagram in Figure 6.1. The implemented repitch-free landmarks were named ‘milestones’ to avoid confusion.

1. A new component `landmark2milestone` converts sequences of extracted landmarks $\{L\}$ to sequences of milestones $\{M\}$ as defined by equation 6.3.
2. The hashing function `landmark2hash` is replaced by `milestone2hash`, implementing equation (6.1).
3. The matching function `get_hash_hits` is adapted to implement equation (6.2).

In addition to this, the implementation provides the possibility of searching for landmarks similar to the ones retrieved. The set of three-dimensional hash components $(f_1, \Delta f, \Delta t)$ of the query can be expanded in one, two or three dimensions by setting the expansion dimension X to 1, 2 or 3. Expansion here means that a deviating copy of the landmark

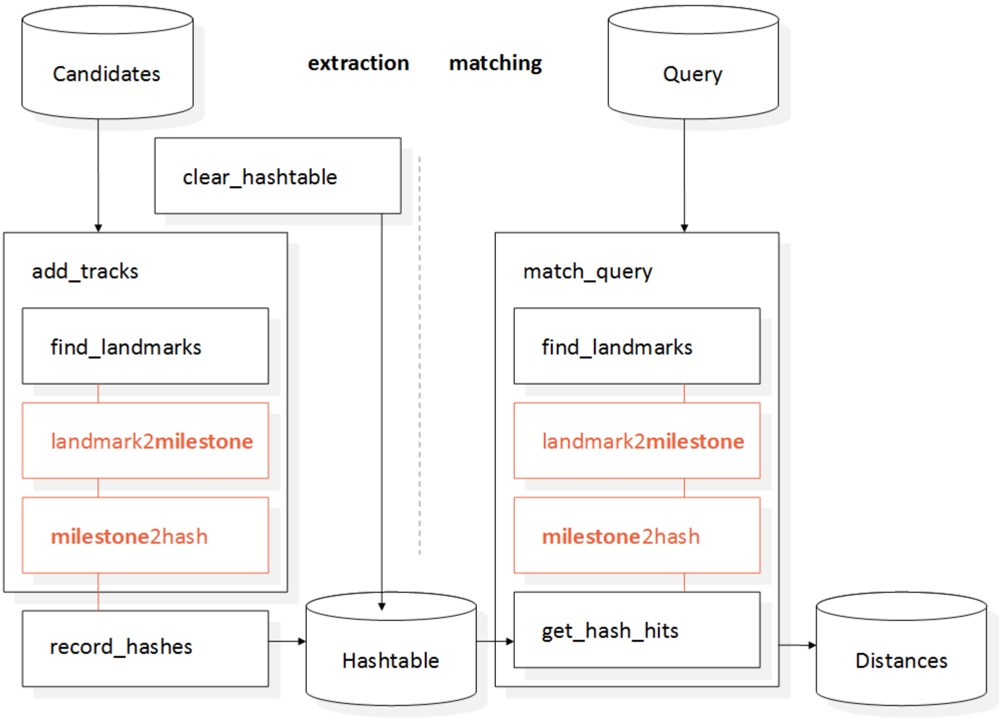


Figure 6.1: Block diagram overview of the adjusted landmark fingerprinting system as described in section 6.1. Each block represents a Matlab function of which the function should be clear by the name. The red blocks are new.

is made for every neighbouring integer value in that dimension. The result is that the landmarks extracted from the query audio will be duplicated 3^X times.

The inclusion of similar hashes in the search process is provided as an attempt to predict the effect of rounding time and frequency values to discrete frames and bins. The way frequency values are rounded to bins (i.e. up or down) might differ before and after repitching.

6.1.2 Results

A series of experiments has been done to assess the feasibility of this repitch invariance, but only the best one will be reported here. For this experiment all parameters except X carry over from the tests with regular landmarks. The repitch-free landmarks were evaluated using the most promising configuration found so far, so sample rate $SR = 2000$ Hz was used. A MAP of 0.218 is achieved with this configuration.

The effect of including similar landmarks is studied next. Table 6.1 lists the results. In the

N_W	H_W	SR	$N(ms)$	$H(ms)$	X	MAP_n	MAP_a
12	6	2000	128	64		0.218	0.218
12	6	2000	128	64	1	0.218	0.198
12	6	2000	128	64	2	0.218	0.174
12	6	2000	128	64	3	0.218	0.145

Table 6.1: Results of experiments with the repitch-free landmarks. In the three last experiments, the extracted landmarks were duplicated 3^X times and varied in an attempt to predict rounding effects.

three last experiments, the extracted landmarks were expanded in the three dimensional 'hash space' by various amounts, choosing $X = 1$, 2 and 3. This inclusion of similar subfingerprints has a negative effect on performance.

6.1.3 Discussion

The highest performance is not far from the 0.228 reference (the MAP obtained with regular landmarks and the same extraction parameters). This is promising, but also expected: the majority of matching landmarks will still match after conversion to repitch-free landmarks or 'milestones'. A more important observation is that no new samples are retrieved. Two new samples that weren't correctly identified by the reference experiment are now retrieved top one, but both of them were identified in similar experiments before, and both are untransposed.

The low performance can have several causes. For example, the selection of peaks and pairs is a complex process and is expected to be similar for similar audio, e.g. sounds that have undergone filtering and the addition of noise. In the case of repitched audio, the spectrum of query and matching candidates differ in a much more nonlinear way. Perhaps the landmark extraction algorithm responds in an equally nonlinear manner, thus extracting radically different landmarks.

However, another flaw in the strategy was found that can be shown to play a major role. For corresponding, repitched landmarks L_A and L_B in a high resolution time-frequency space, the products $(\Delta f_A \cdot \Delta t_A)$ and $(\Delta f_B \cdot \Delta t_B)$ can differ slightly because of the rounding of times and frequencies to frames and bins. Though in a spectrum where the resolutions are rather low, as is the case, their difference is significant. The prediction of rounding errors as provided with parameter X is far not enough to account for the observed deviations.

Accuracy in the Product Space

As an example, consider the following landmark extracted 76.8 seconds into a song C, with $H = 64$ ms and time and frequency constraints $\Delta t_{max} = 63$ and $\Delta t_{max} = 31$.

$$L_C = \{1200,\ 25,\ 40,\ 45\}.$$

The products in the hash would then become

$$\begin{array}{rcccccl} P_{1C} & = & f_{1C} \cdot \Delta t_C & = & 25 \cdot 45 & = & 1125 \\ P_{2C} & = & \Delta f_C \cdot \Delta t_C & = & 15 \cdot 45 & = & 675 \end{array}$$

Suppose the same landmark appears in the query Q that samples song C, but the sample has been repitched 2 semitones or 12.25% up.

$$\begin{array}{rcccl} f_{1Q} & = & [1.1225 \cdot 25] & = & 28 \\ f_{2Q} & = & [1.1225 \cdot 40] & = & 45 \\ \Delta t_Q & = & [45 \,/\, 1.1225] & = & 40 \\ \Delta f_Q & = & 45 - 28 & = & 17 \end{array}$$

$$L_C = \{80,\ 25,\ 28,\ 40\}.$$

Where the square brackets denote rounding to the nearest integer. The products in the hash for the query would then become

$$\begin{array}{rcccccl} P_{1Q} & = & f_{1Q} \cdot \Delta t_Q & = & 28 \cdot 40 & = & 1120 \\ P_{2Q} & = & \Delta f_Q \cdot \Delta t_Q & = & 17 \cdot 40 & = & 680 \end{array}$$

The difference between candidate and query is 5 'framebins' in every product, or a deviation of 0.45% and 0.74%, respectively. In order to be able to include these deviations in the same hash, a tolerance on the order of 5 framebins would be needed. This may not seem much, but it means the entropy in each product would be 5 times lower. Moreover, it can be shown that for the majority of landmarks with this resolution, a margin of 5 would not even be enough. A calculation over a large and uniform sample of possible landmarks and repitch factors (up to 3 semitones) shows a median difference of 12 and 6, respectively.

Locality Sensitive Hashing

One could implement a simple form of Locality Sensitive Hashing (see section 2.3.1) to account for these deviations, for example by partitioning the 'two products' space $P_1 \times P_2$ into 12 × 6 boxes. For a maximum P_1 and P_2 of 8192 and 4096, this would yield less than 500,000 possible hashes, and that does not even account for the redundancy in Δt (as it is used twice). A logarithmic partitioning of the two products space based on median percentual deviations rather than differences, gives more or less the same number of possible hashes. The observed deviations of 0.8% and 1.8% require the number of logarithmically spaced boxes to be lower than 530,000. Coming from the 33 million hash combinations in the previous experiment, this corresponds to a loss of 6 bits of precious entropy to preserve only half of the landmark (as the median was used to assess deviations).

Solutions to this issue may exist. Perhaps settling with a lower entropy and moving some of the discriminative power to the matching of τ, exploiting correlation of deviations between the two products, or designing another LSH strategy could solve some of the problem. Concerning the research done for this report, it has been decided that this strategy is not promising enough to be investigated further. The next experiments focus on the implementation of a more pragmatic method to search for repitches.

6.2 Repitching Landmarks

An attempt to design repitch-invariant landmark-like subfingerprints has not been succesful. One method to search for repitched audio that is very straightforward and has been left aside up to this point, consists of repitching the query audio for several repitch amounts R and computing a combined distance matrix containing for every (Q, C) pair the lowest distance over all R. This strategy will be explored in the present section. First, however, a pragmatic method is proposed to handle small repitch amounts without having to repitch the audio itself. This way, a set of repitch factors can be scanned through without systematically overlooking all values in between.

6.2.1 Methodology

Repitching landmarks

Small repitch amounts may be dealt with by predicting the way a landmark will change under repitching with a certain factor r. Deviating landmarks predicted from those extracted from the query are included in the database matching. The following very straightforward computation shows how predictions can be calculated from an extracted landmark L_Q for a series of small repitch factors $\{r\}$ (for example all 1 cent apart[1]). For every r:

$$\begin{array}{rcl} f_{1r} & = & [r \cdot f_{1Q}] \\ f_{2r} & = & [r \cdot f_{2Q}] \\ \Delta t_r & = & [\Delta t_Q \,/\, r] \end{array} \tag{6.4}$$

However, if the spectrogram from which the landmarks are extracted is computed using the constant Q transform, which features logarithmically spaced frequencies k, the mapping in equation 6.4 changes to:

$$\begin{array}{rcl} k_{1r} & = & [f_{1Q} + \log_2(r) \cdot bpo] \\ k_{2r} & = & [f_{2Q} + \log_2(r) \cdot bpo] \\ \Delta t_r & = & [\Delta t_Q \,/\, r] \end{array} \tag{6.5}$$

[1] A cent is 1/100 of a semitone, or a relative frequency difference of 0.06%

where *bpo* was the constant Q transform's number of bins per octave.

As a consequence, the landmark component $\Delta k = k_2 - k_1$ is always unchanged under this transformation. This is an advantage: it reduces the number of hash dimensions in which the landmarks are expanded, so that less deviating copies of the query's landmarks have to be made.

The predictions described above are calculated in the Matlab function `tool_stretch`, which is newly implemented and called by `match_query`. Its parameters are r_{min} and r_{max} and, given a set of extracted landmarks, it returns all unique predictions (no exact duplicates).

Repitching query audio

Larger amounts of repitch will be dealt with by repitching the query audio several times and performing a complete experiment involving all queries for each of these repitches. The repitching of the audio is performed by a new function `repitch` calling Matlab's `resample` algorithm. This requires the resample amount R in semitones (st) to be translated to an integer upsample and downsample factor P and Q, where

$$\log_2(R/12) \approx Q/P$$

as there are 12 semitones in an octave, and repitching audio up requires a lower samplerate. The optimal factors P and Q are found using Matlab's `rat` function with precision 0.006, or one cent.

In the results below, parameters R_{min}, R_{max} and ΔR are used to describe how many experiments are performed. Within every experiment, r_{min} and r_{max} can then be chosen to cover the repitch factors between the R. Note that complete coverage requires

$$r_{max} - r_{min} \;\geq\; \log_2(\Delta R/12).$$

As said in the beginning of this section, the distance matrices returned by every experiment are combined by taking for every (Q, C) pair the lowest distance over all R. From the resulting matrix, the MAP is computed to assess performance.

6.2.2 Results

A first experiment features no repitching of the audio itself ($R = 0$ st). Though with $r_{min} = 0.972$ and $r_{max} = 1.029$, or half a semitone up and down, the performance of the constant Q-based landmark system could be increased from $MAP = 0.211$ to 0.288, a very significant improvement.

A second set of experiments covered repitches from -2 to $+2$ semitones with a step of $\Delta R = 1$ semitone. To have complete coverage, (r_{min}, r_{max}) was again chosen $(0.972, 1.029)$. The obtained MAP is a best-so-far 0.341.

SR	R_{min}	R_{max}	ΔR	r_{min}	r_{max}	MAP_n	MAP_a
8000	0	0	0	0	0	0.211	0.170
8000	0	0	0	-0.5	0.5	0.268	0.288
8000	-2	+2	1.0	-0.5	0.5	0.341	0.334
8000	-2	+2	0.5	-0.5	-0.5	0.373	0.390

Table 6.2: Results of experiments using repitching of both the query audio and its extracted landmarks to search for repithed samples.

	Drum samples	Tonal samples	Total
Repitched samples	5 (1)	3 (0)	8 (1)
Non repitched samples	4 (4)	17 (9)	21 (13)
Total	9 (5)	20 (9)	29 (14)

Table 6.3: Sample type statistics for the 29 correct matches retrieved by the best performing system and the 14 correct matches of a reference performance achieved using untransposed constant Q-based landmarks (in parentheses).

The third set of experiments involved some overlap in the coverage: ΔR was halved to 0.5 semitones, adding a number of new experiments to the previous set, with the same (r_{min}, r_{max}) but non-integer R. The results were better ($MAP = 0.391$) at the cost of doubling the total computation time.

Experiments with repitched queries take several hours for every R, so the number of experiments that could be done was limited. Details for all three series of experiments are given in Table 6.2. These results are now discussed.

6.2.3 Discussion

The best performing system achieved MAPs of 0.373 and 0.390. A record 29 of the 76 queries retrieve a correct first match. This is indeed a great improvement over the previous best system, which reached MAPs of 0.211 and 0.170.

Every of the 29 correct best retrievals was now checked to see if it was a drum sample or a tonal sample (which includes the drum samples with tonal elements) and whether or not its was repitched (or perhaps time-stretched in the case of drum samples, for which the difference is difficult to perceive). Table 6.3 lists these data. The numbers in the parentheses are for the reference experiment involving untransposed constant Q-based landmarks (achieving the MAP of 0.211).

Even with 29 matching queries, the conclusions drawn from these statistics will be relative. The general statistics of the database that would be needed to situate these numbers are unknown, especially when it comes to drum samples. However, it is safe to say that:

1. An additional number of unrepitched, tonal samples were retrieved, even though the

experiment was not set up for this. This could be a result of the multiple searches with each query, or of the inclusion of deviating landmarks, amongst many others.

2. Both repitched samples, and unpitched drum samples have now been successfully recognised, thus accomplishing the main goal of the Chapters 5 and 6.

3. It remains hard to tell if the degree to which drum samples could be fingerprinted meets the requirement an automatic hip hop sample detection system should meet.

Despite the last point, the MAP of almost 0.4 is remarkable for a first attempt at the proposed task of automatic sample recognition.

Chapter 7

Discussion and Future Work

In this last chapter, findings and results from all chapters are summarised. This leads to some conclusions and prospects for possible future work.

7.1 Discussion

7.1.1 Contributions

This is the first research known to address the problem of automatic sample identification. As a result, a summary of this thesis' contributions can include the problem statement itself, though not without a thorough overview of the particularities of sampling as listed in Chapter 1. To summarise, this thesis has achieved the following objectives.

1. The problem of sample identification as a music information retrieval task has been defined and situated in the broader context of sampling as a musical phenomenon. This includes listing the requirements a sample recognition system must meet:

    - Given a music collection, the system should be able to identify known, but heavily manipulated query audio.
    - The system should be able to do this for large collections (e.g. over 1000 files).
    - The system should be able to do this in a reasonable amount of time (e.g. up to several hours).

2. The most relevant research to date has been brought together and critically reviewed in terms of these requirements. The main challenge amongst the above requirements has been identified as:

    - Dealing with timestretching and transposition of samples
    - Dealing with non-tonal samples

3. A representative collection of hip hop sampling examples has been assembled and annotated with relevant metadata. This dataset, together with a number of proposed evaluation metrics and their random baselines described in Chapter 3, is a valuable contribution that can definitely be re-used in future work on the topic. The random baseline of the MAP evaluation metric was found to be 0.017 (over 100 iterations).

4. A promising state-of-the-art audio fingerprinting algorithm has been optimised and tested in Chapter 4, to set a state-of-the-art baseline for the evaluation. It has been found to perform poorly, but several times better than the random baseline: $MAP_n = 0.147$, $MAP_n = 0.128$.

5. A selection of possible approaches to automatic sample identification has been investigated in Chapters 5 and 6. The most promising approach performed several times better than the state-of-the-art algorithm: $MAP_n = 0.373$, $MAP_n = 0.390$. This is a remarkable performance for a first attempt at a difficult and newly proposed task.

The best achieved performance is obtained using the last of the proposed strategies. It involves the extraction and matching of landmarks as proposed by Wang [3], but defines a new type of landmarks, and adopts a combined strategy of audio repitching and fingerprint variation to obtain a certain degree of robustness to repitching. The most important findings are the following:

- Exploiting the interdependence of time-stretching and transposition in repitched audio segments to represent them in a repitch-independent way requires a large resolution of the spectrogram. The idea as it was implemented could not provide enough entropy from realistic time and frequency resolutions.

- Lowering the sample rate at which the audio is analysed, and using a constant Q transform to obtain a log frequency spectrogram of the processed audio segments (instead of a linear frequency spectrogram) were both found to increase the system's performance.

- Calculating repitched versions of extracted landmarks allows for retrieval of slightly repitched samples. Meanwhile, repitching query audio several times proved complementary to this repitching of extracted landmarks and allows for the retrieval of samples featuring a broader range of repitch amounts.

7.1.2 Error Analysis

As explained in section 6.2.3, an exhaustive error analysis is difficult to perform because several required statistics were not annotated. Every sample indicates if a beat or a riff was sampled, but for a critical look at retrievals, this is not enough; beats may still contain significant tonal elements. In the same way, there is no information on how many of the beats appear isolated. Finally, the amount of time stretching and transposition of samples is not annotated and often difficult to assess. As a result, no conclusions can be made on why the non-recognised samples are not retrieved. However, a few hypotheses can be stated:

- The samples that have not been retrieved appear underneath many layers of other musical elements. It has not been observed that all retrieved samples were especially isolated or high in the mix.
- No strategy has yet been considered to detect samples that were times-stretched and transposed independently. At least a few of the samples in the database feature this kind of transformation.
- For some files, more landmarks were extracted than for others. This has been observed but never closely studied. Perhaps the amount of extracted samples made some files more difficult to retrieve.

7.1.3 Critical Remarks

An acceptable degree of performance has been reached, considering this is the first time the task is addressed. Yet, the results could have been better. A number of critical remarks can be made regarding the extent of the contributions listed above.

1. Only less than half of the queries in the database retrieved a correct candidate first. A large number of samples are still unidentified, and no particular reason for this stands out. Especially the system's performance recognising drum samples is difficult to judge.
2. Not all parameters of the system could be optimised, because of constraint in both time and computational resources. Even the parameters that have been optimised in Chapter 4 are perhaps far from optimal in the modified system of Chapter 6. The performance of the system proposed last may be increased significantly if another optimisation were performed.
3. The collection that was used to evaluate was limited. Any real-life music collection that would be used in a computational ethnomusicological study of sampling would exceed the used collection several times in size. The discriminative power may decrease in such a context, and noise matches from unrelated files may play a larger role.

7.2 Future Work

The research done in this thesis is mostly an exploration of a selection of strategies to address the newly defined problem of automatic sample detection. One big question has been asked, but only small answers are given. Many more approaches could and should be investigated.

Within the problem of sample identification, particularly interesting issues that may be further addressed are:

- Can a system be implemented to perform the proposed task with a accuracies comparable to previously achieved in audio fingerprinting and currently achieved in cover detection?
- How can percussive sounds be reliably fingerprinted? The challenges that come with the fingerprinting of these sounds may be of a whole other kind than those emphasised on in this thesis.
- Are there other types of samples that are particularly challenging to identify? An extension of the annotations in the music collection could help identify more specific problems within the task.
- The identification of independently time-stretched and transposed samples. No strategy that could do this has been presented so far.

On a bigger scale, it could be interesting to:

- Investigate if the way an automatic system could identify samples, reflects any of the mechanisms humans use to recognise familiar music excerpts. Do parallels with music cognition exist?
- Identify and analyse common properties of the different styles and types of fragments that have been sampled. Is there such a thing as the sampleability of a piece of music?
- Embed the identification of sampled audio in a larger study of re-use of musical ideas. How do musical ideas propagate through time, and does sampling contribute to this?

Hopefully this thesis can be a part of a new interesting line of research within Music Information Retrieval.

Appendix A

Derivation of τ

Prior to the repitch, the time offset between candidate and query τ was defined as

$$\tau = t_{1C} - t_1.$$

This relation is now broken by the introduction of a repitch factor r scaling the candidate's timestamp t_{1C}:

$$\tau_C = r \cdot t_{1C} - t_1.$$

The same goes for the candidate time difference Δt_C and frequency difference Δf_C.

$$\begin{aligned} \Delta t &= r \cdot \Delta t_C \\ \Delta f &= \Delta f_C \,/\, r. \end{aligned}$$

To obtain τ, the repitch factor r can then be obtained by substitution of r

$$\tau = t_{1C} \cdot \frac{\Delta t}{\Delta t_C} - t_1.$$

The same histogram plots as in Figure 2.7 can be used to find the winning offset.

Appendix B

Music Collection

	Artist	Title	Year	Genre
T001	OC	Time's Up	1994	Hip-hop
T002	Les Demerle	A Day in the Life	1968	Jazz
T003	David Axelrod	Holy Thursday	1968	Jazz
T004	Lil Wayne	Dr. Carter	2008	Hip-hop
T005	Clyde McPhatter	Mixed Up Cup	1970	R&B/Soul
T006	Common	In My Own World (Check the Method)	1994	Hip-hop
T007	Monty Alexander	Love and Happiness	1974	Jazz
T008	The Beatnuts	Let Off a Couple	1994	Hip-hop
T009	Beastie Boys	Rhymin & Stealin'	1986	Hip-hop
T010	Led Zeppelin	When The Levee Breaks	1971	Rock
T011	Bill Withers	Grandma's Hands	1971	R&B/Soul
T012	Blackstreet	No Diggity	1996	Hip-hop
T013	Jay-Z	Roc Boys (and the Winner is)	2007	Hip-hop
T014	Menahan Street Band	Make the Road by Walking	2006	World
T015	Lou Reed	Walk on the Wild Side	1971	Rock
T016	A Tribe Called Quest	Can I Kick it?	1990	Hip-hop
T017	House of Pain	Jump Around	1992	Hip-hop
T018	Bob & Earl	Harlem Shuffle	1963	R&B/Soul
T020	Will Smith	Miami	1997	Hip-hop
T021	The Whispers	And the Beat Goes On	1980	Disco
T022	Asheru ft. Talib Kweli	Mood Swing	2002	Hip-hop
T023	Duke Ellington and John Coltrane	In a Sentimental Mood	1963	Jazz
T024	Wu-Tang Clan	C.R.E.A.M.	1993	Hip-hop
T025	The Charmels	As Long as I've Got You	1967	R&B/Soul
T026	Beastie Boys	Rock Hard	1985	Hip-hop
T027	AC/DC	Back in Black	1980	Rock
T028	Wu-Tang Clan	Tearz	1993	Hip-hop
T029	Wendy Rene	After Laughter (Comes Tears)	1964	R&B/Soul
T030	Non Phixion	Rock Stars	2002	Hip-hop
T031	Bar-Kays	In the hole	1969	R&B/Soul
T032	A Tribe Called Quest ft. Leaders of the New School & Kid Hood	Scenario (Remix)	1992	Hip-hop
T033	The Emotions	Blind Alley	1972	R&B/Soul
T034	Pete Rock & C.L. Smooth	Straighten it Out	1992	Hip-hop
T035	Ernie Hines	Our Generation	1972	R&B/Soul
T036	Big Daddy Kane	Ain't No Half-Steppin'	1988	Hip-hop
T037	De La Soul	Eye Know	1989	Hip-hop
T038	The Mad Lads	Make This Young Lady Mine	1969	R&B/Soul
T039	Biz Markie	Just a Friend	1989	Hip-hop
T040	Freddie Scott	You Got What I Need	1968	R&B/Soul
T041	Killah Priest ft. Hell Razah and Tekitha	One Step	1998	Hip-hop

	Artist	Title	Year	Genre
T041	Killah Priest ft. Hell Razah and Tekitha	One Step	1998	Hip-hop
T042	William Bell	I Forgot to Be Your Lover	1968	R&B/Soul
T043	Otis Redding	(Sittin' On) the Dock of the Bay	1968	R&B/Soul
T044	Wu-Tang Clan	Protect Ya Neck	1993	Hip-hop
T045	The J.B.'s	The Grunt	1970	R&B/Soul
T048	De La Soul	Change in Speak	1989	Hip-hop
T049	Cymande	Bra	1972	Funk
T050	The Mad Lads	No Strings Attached	1969	Hip-hop
T051	Nas	Memory Lane (Sittin' in Da Park)	1994	Hip-hop
T052	Lee Dorsey	Get Out of My Life, Woman	1966	R&B/Soul
T053	Fat Joe	Flow Joe	1993	Hip-hop
T054	Public Enemy	Rebel Without a Pause	1987	Hip-hop
T056	Blowfly	Outro	1973	Funk
T057	Jurassic 5	Quality Control	2000	Hip-hop
T058	David McCallum	The Edge	1967	Classical
T059	Dr. Dre ft. Snoop Dogg	The Next Episode	1999	Hip-hop
T060	Bobby Caldwell	Open Your Eyes	1980	Jazz
T061	Common	The Light	2000	Hip-hop
T062	Dido	Thank You	2000	Pop
T063	Eminem	Stan	2000	Hip-hop
T064	Eddie Holman	It's Over	1977	R&B/Soul
T065	Ghostface Killah & RZA	Nutmeg	2000	Hip-hop
T066	Foreigner	Cold as Ice	1977	Rock
T067	M.O.P.	Cold as Ice	2000	Hip-hop
T068	Grace Jones	Nightclubbing	1981	Disco
T069	Shyne & Barrington Levy	Bad Boyz	2000	Hip-hop
T070	Hossam Ramzy	Khusara Khusara	1994	World
T071	Jay-Z & UKG	Big Pimpin'	1999	Hip-hop
T072	Jack Mayborn	Music People	1978	R&B/Soul
T073	Prodigy	Keep it Thoro	2000	Hip-hop
T074	Jimmie & Vella Cameron	Hey Boy Over There	1968	R&B/Soul
T075	Capone N' Noreaga	Invincible	2004	Hip-hop
T076	Raymond Lefevre & His Orchestra	The Days of Pearly Spencer	1967	R&B/Soul
T077	Black Rob	Whoa!	2000	Hip-hop
T078	Sam & Dave	Soul Sister, Brown Sugar	1969	R&B/Soul
T079	M.O.P.	Ante Up	2000	Hip-hop
T080	Solomon Burke	Cool Breeze	1972	R&B/Soul
T081	Ghostface Killah & Raekwon	Apollo Kids	2000	Hip-hop
T082	The Monkeys	Mary Mary	1966	Rock
T083	James Brown	Funky Drummer	1970	Funk

	Artist	Title	Year	Genre
T084	Jay-Z ft. Alicia Keys	Empire State of Mind	2009	Hip-hop
T085	Isaac Hayes	The Breakthrough	1974	R&B/Soul
T086	NWA	Straight outta Compton	1988	Hip-hop
T088	Funkadelic	You'll like it too	1981	Funk
T090	The Honey Drippers	Impeach The President	1973	R&B/Soul
T091	Nice & Smooth	Funky For You	1989	Hip-hop
T092	LL Cool J	Around the Way Girl	1990	Hip-hop
T093	Kris Kross	Jump	1992	Hip-hop
T094	Jackson 5	I want you back	1969	R&B/Soul
T095	Ohio Players	Funky Worm	1972	R&B/Soul
T097	Naughty by Nature	Hip Hop Hurray	1993	Hip-hop
T098	Five Stairsteps	Don't Change Your Love	1968	R&B/Soul
T099	A Tribe Called Quest	Jazz (We've Got)	1991	Hip-hop
T100	Ice Cube	A Bird in the Hand	1991	Hip-hop
T101	Southside Movement	I been watchin' you	1973	Funk
T102	Cormega	American Beauty	2001	Hip-hop
T103	The Avalanches	Since I left you	2000	Hip-hop
T104	Lamont Dozier	Take off your make up	1973	R&B/Soul
T108	Gary Numan	Films	1979	Rock
T109	DJ Qbert	Eight	1994	Hip-hop
T110	Ghostface Killah & RZA	The Grain	2000	Hip-hop
T111	Rufus Thomas	Do the Funky Penguin	1971	R&B/Soul
T112	Rufus Thomas	The Breakdown (part 2)	1971	R&B/Soul
T115	Salt-N-pepa	I desire	1986	Hip-hop
T116	The Winstons	Amen, Brother	1969	R&B/Soul
T117	7th Wonder	Daisy Lady	1979	R&B/Soul
T118	Kanye West ft. Nas, Really Doe	We Major	2005	Hip-hop
T119	Orange Krush	Action	1982	Funk
T121	LL Cool J	Breakthrough	1987	Hip-hop
T144	Beastie Boys	Time to get Ill	1986	Hip-hop
T145	Barry White	I'm gonna love you just a little more baby	1973	R&B/Soul
T146	De La Soul	De la Orgee	1989	Hip-hop
T147	Ras Kass	Rasassination	1998	Hip-hop
T148	Johnny Pate	Shaft in Africa	1973	R&B/Soul
T149	Jay-Z	Show me what you got	2006	Hip-hop
T150	Cypress Hill	Real Estate	1991	Hip-hop
T151	All The People	Cramp Your Style	1972	R&B/Soul
T152	Nas	One Mic	2001	Hip-hop
T153	Public Enemy	911 is a Joke	1990	Hip-hop
T154	Sound Experience	Devil with the bust	1974	R&B/Soul

	Artist	Title	Year	Genre
T157	Geto Boys	Fuck a War	1991	Hip-hop
T160	Bobby Byrd	I Know You Got Soul	1971	R&B/Soul
T162	Joe Tex	Papa Was Too	1966	R&B/Soul
T163	Wu-Tang Clan	Wu-Tang Clan ain't nuthin ta F' wit	1993	Hip-hop
T164	EPMD	Jane	1988	Hip-hop
T166	Nikki D	Lettin' Off Steam	1990	Hip-hop
T167	The Politicians	Free Your Mind	1972	R&B/Soul
T172	3rd Bass	Oval Office	1989	Hip-hop
T173	Joe Quarterman & Free Soul	I'm gonna get you	1974	R&B/Soul
T176	Beastie Boys	Egg man	1989	Hip-hop
T177	Curtis Mayfield	Superfly	1972	R&B/Soul
T178	Crucial conflict	Showdown	1996	Hip-hop
T179	Beastie Boys	Brass Monkey	1986	Hip-hop
T180	Wild Sugar	Bring it Here	1981	Disco
T181	The Notorious B.I.G.	Respect	1994	Hip-hop
T182	KC & The Sunshine Band	I Get Lifted	1975	R&B/Soul
T184	Nas	Get Down	2002	Hip-hop
T185	The Blackbyrds	Rock Creek Park	1975	R&B/Soul
T187	Erik B. & Rakim	I Know You Got Soul	1987	Hip-hop
T189	NWA	Fuck the Police	1988	Hip-hop
T190	Erik B. & Rakim	Lyrics of Fury	1988	Hip-hop
T191	LL Cool J	Mama said knock you out	1990	Hip-hop
T192	Public Enemy	Welcome to the Terrordome	1989	Hip-hop
T193	Dyke & The Blazers	Let a woman be a woman, let a man be a man	1969	Funk
T199	James Brown	The Boss	1973	R&B/Soul

Table B.1: All tracks in the database.

	C	Q	TC	TQ	N	Comments
S001	T002	T001	0:00	0:00	19	chopped (sample ABCD looped ABCBCBCD)
S002	T003	T004	0:00	0:00	4	intro
S003	T003	T004	0:34	0:53	3	refrain with horns
S004	T005	T006	0:00	0:37	60	
S005	T007	T008	0:53	0:00	10	(very clean: same pitch and only layer)
S006	T010	T009	0:00	0:01	37	probably samples both bars (with two decks?)
S007	T011	T012	0:01	0:00	85	(sample a bit gated in the end)
S008	T014	T013	1:14	0:05	10	beginning of riff counted (rest 'chopped')
S009	T015	T016	0:00	0:03	37	interpolation or other version
S010	T018	T017	0:00	0:00	1	
S012	T021	T020	0:00	0:00	38	time sampled unclear (0:00 or 2:25?)
S013	T023	T022	0:01	0:10	51	piano
S014	T025	T024	0:00	0:21	37	
S015	T027	T026	0:05	0:12	38	
S016	T029	T028	0:00	0:37	50	
S017	T031	T030	0:08	0:10	43	first half of loop
S018	T033	T032	0:01	0:18	217	(samples 2nd measure, hence short and no melody)
S019	T035	T034	0:40	0:10	48	(vocals)
S020	T033	T036	0:00	0:02	33	
S021	T038	T037	0:00	0:00	50	the guitar bar
S022	T043	T037	2:21	0:26	22	
S023	T040	T039	0:00	0:02	21	interpolation
S024	T042	T041	0:00	0:04	40	the guitar bar
S025	T045	T044	0:00	0:39	67	very soft and hard to count
S027	T049	T048	0:00	0:09	25	
S028	T050	T048	0:00	0:00	19	half samples not counted (second halves)
S029	T052	T051	0:00	0:10	85	only the measures with single BD counted
S030	T052	T039	0:00	0:13	76	
S031	T052	T037	0:00	0:04	110	BD filtered out or replay
S032	T052	T053	0:01	0:20	147	HH filtered out or replay
S033	T045	T054	0:00	0:12	92	
S035	T056	T057	0:03	0:06	44	
S036	T058	T059	0:05	0:10	26	Short sample intro not counted
S037	T060	T061	1:03	1:03	3	vocals only
S038	T062	T063	0:35	0:01	5	refrain w/ vocals (long)
S039	T064	T065	0:25	0:22	157	the flute, verse
S040	T066	T067	0:07	0:18	34	both instr/vocal, instr parts chopped?
S041	T068	T069	0:06	0:13	23	nice and long (11 seconds)
S042	T070	T071	0:00	0:00	38	
S043	T072	T073	0:00	0:05	55	piano part

	C	Q	TC	TQ	N	Comments
S044	T074	T075	0:07	0:08	79	Short, alternates full/half sample
S045	T076	T077	0:32	0:00	143	Only 8 note string riff counted
S046	T078	T079	0:04	0:02	151	Less than one second
S047	T080	T081	0:03	0:20	15	Only repetitions of string riff counted
S048	T082	T048	0:02	0:00	68	
S049	T083	T054	5:21	0:12	80	
S050	T085	T084	0:01	0:00	45	
S051	T116	T086	1:28	0:20	80	
S052	T088	T086	0:01	2:59	3	
S054	T090	T091	0:00	0:03	91	
S055	T090	T092	0:00	0:00	101	
S056	T094	T093	0:05	0:00	60	
S057	T090	T093	0:00	0:00	150	half of the samples used in S054 and S55
S058	T095	T093	2:18	0:00	40	
S060	T098	T097	0:00	0:08	204	pretty short but ok
S061	T098	T099	0:00	0:11	150	Pretty short, only full cycles counted
S062	T098	T100	0:00	0:05	100	pretty short but ok
S063	T101	T102	0:01	0:00	24	
S064	T104	T103	0:01	0:15	96	
S067	T108	T109	0:06	0:54	114	
S068	T111	T110	0:00	0:15	55	
S069	T112	T110	0:14	0:00	1	
S071	T116	T115	1:28	0:07	82	
S072	T117	T115	0:01	1:00	7	
S073	T119	T118	0:05	0:00	139	
S074	T085	T121	0:01	0:00	74	Perhaps chopped with compression or gated reverb: different timbre; Yet accurate pattern
S078	T145	T144	0:12	0:39	2	
S079	T145	T146	0:01	0:00	22	
S080	T148	T147	0:18	0:05	24	
S081	T148	T149	0:18	0:13	15	end of loop doubled (not counted)
S082	T151	T150	0:06	1:54	7	
S083	T145	T152	0:03	0:56	28	
S084	T154	T153	0:18	0:00	54	LF only
S086	T154	T157	0:01	0:01	9	intro
S089	T160	T157	0:00	1:04	8	
S091	T162	T163	0:00	0:17	3	HF only and soft
S092	T162	T164	0:08	0:07	53	
S094	T167	T166	0:06	0:15	43	Bass + synth
S098	T173	T172	1:48	0:05	71	
S101	T177	T176	0:00	0:00	27	

	C	Q	TC	TQ	N	Comments
S102	T177	T178	0:00	0:07	70	
S103	T180	T179	0:01	0:01	12	
S104	T182	T181	0:00	0:00	99	
S106	T185	T184	0:00	2:15	8	
S107	T083	T184	5:21	2:17	7	
S109	T160	T187	0:00	0:09	77	might be 39
S112	T088	T187	0:01	0:00	21	
S113	T083	T189	5:21	2:17	4	might be 2
S114	T083	T190	5:21	0:00	102	
S115	T083	T191	5:21	0:11	112	
S116	T193	T192	1:46	0:05	75	Beginnings counted
S120	T199	T184	0:05	0:02	71	
S121	T064	T065	0:02	0:00	12	the intro/refrain riff
S122	T072	T073	0:04	0:32	5	brass part
S123	T062	T063	0:59	0:49	44	verse (very soft)
S124	T052	T051	0:01	0:13	80	only the measures with double BD counted
S125	T042	T041	0:07	0:12	16	the guitar + strings bar
S126	T038	T037	0:04	0:52	10	the guitar + brass bar
S127	T031	T030	0:46	0:13	43	end of loop
S128	T029	T028	0:05	0:27	4	10 sec refrain
S129	T023	T022	0:11	1:24	2	saxophone
S130	T014	T013	1:08	0:00	1	intro not interpolated
S131	T003	T004	0:23	0:20	9	repeated beat (of which 3 included in S2)
S134	T167	T166	0:02	0:25	14	just bass
S135	T199	T184	0:14	0:58	3	brass riff
S136	T199	T184	1:06	0:00	2	brass crescendo
S137	T173	T172	0:01	0:44	14	

Table B.2: All samples in database

Bibliography

[1] M. Casey, C. Rhodes, and M. Slaney, "Analysis of Minimum Distances in High-Dimensional Musical Spaces," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 1015–1028, July 2008.

[2] P. Cano, E. Battle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," *2002 IEEE Workshop on Multimedia Signal Processing.*, pp. 169–173, 2002.

[3] A. L.-C. Wang, "An industrial strength audio search algorithm," in *International Conference on Music Information Retrieval (ISMIR)*, 2003.

[4] D. P. W. Ellis, "Robust Landmark-Based Audio Fingerprinting," *http://labrosa.ee.columbia.edu/matlab/fingerprint/*, 2009.

[5] J. Herre, E. Allamanche, and O. Hellmuth, "Robust matching of audio signals using spectral flatness features," *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, no. October, pp. 127–130, 2001.

[6] J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprinting System With an Efficient Search Strategy," *Journal of New Music Research*, vol. 32, pp. 211–221, June 2003.

[7] T. McKenna, "Where Digital Music Technology and Law Collide - Contemporary Issues of Digital Sampling, Appropriation and Copyright Law," *Journal of Information Law and Technology*, vol. 1, pp. 0–1, 2000.

[8] D. Hesmondhalgh, "Digital Sampling and Cultural Inequality," *Social & Legal Studies*, vol. 15, no. 1, pp. 53–75, 2006.

[9] Billboard, "Music News, Reviews, Articles, Information, News Online & Free Music," *www.billboard.com*, 2011.

[10] Wikipedia, "The R.E.D. Album," *Wikipedia - The Free Encyclopedia. Retrieved September 2, 2011; http://en.wikipedia.org/wiki/The_R.E.D._Album.*

[11] Wikipedia, "Dub music," *Wikipedia - The Free Encyclopedia. Retrieved September 12, 2011; http://en.wikipedia.org/wiki/Dub_music.*

[12] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-Based Music Information Retrieval: Current Directions and Future Challenges," *Proceedings of the IEEE*, vol. 96, pp. 668–696, Apr. 2008.

[13] M. Müller, D. Ellis, A. Klapuri, and G. Richard, "Signal Processing for Music Analysis," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 0, no. 99, pp. 1–1, 2011.

[14] J. Demers, "Sampling the 1970s in hip-hop," *Popular Music*, vol. 22, pp. 41–56, Mar. 2003.

[15] A. Volk, J. Garbers, P. Kranenburg, F. Wiering, L. Grijp, and R. Veltkamp, "Comparing Computational Approaches to Rhythmic and Melodic Similarity in Folksong Research," *Mathematics and Computation in Music*, pp. 78–87, 2009.

[16] J. Serrà, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: background, approaches, evaluation, and beyond," in *Advances in Music Information Retrieval*, pp. 307–332, Springer, 2010.

[17] Wikipedia, "Pump It," *Wikipedia - The Free Encyclopedia. Retrieved September 2, 2011; http://en.wikipedia.org/wiki/Pump_It.*

[18] M. Marolt, "A Mid-Level Representation for Melody-Based Retrieval in Audio Collections," *IEEE Transactions on Multimedia*, vol. 10, pp. 1617–1625, Dec. 2008.

[19] W. Fulford-Jones, "Sampling," *Grove Music Online. Oxford Music Online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/47228*, 2011.

[20] W. Marshall, "Kool Herc," in *Icons of Hip Hop: An Encyclopedia of the Movement, Music, and Culture* (M. Hess, ed.), vol. 22, p. 683, 2007.

[21] H. Self, "Digital Sampling: A Cultural Perspective," *UCLA Ent. L. Rev.*, vol. 9, p. 347, 2001.

[22] I. Peel, "Dance Music," *Grove Music Online. Oxford Music Online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/47215*, 2011.

[23] S. Emmerson and D. Smalley, "Electro-acoustic music.," *Grove Music Online. Oxford Music Online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/08695*, 2011.

[24] J. Pritchett and L. Kuhn, "John Cage," *Grove Music Online. Oxford Music Online. http://www.oxfordmusiconline.com/subscriber/article/grove/music/49908*, 2011.

[25] J. Laroche and M. Dolson, "Improved Phase Vocoder Time-Scale Modification of Audio," *Ieee Transactions On Audio*, vol. 7, no. 3, pp. 1–10, 1999.

[26] A. Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFx03)*, pp. 344–349, Citeseer, 2003.

[27] D. Sanjek, "''Don't Have To DJ No More'': Sampling and the ''Autonomous'' Creator," *Cardozo Arts & Ent. L.J.*, vol. 10:607, 1992.

[28] T. Porcello, "The ethics of digital audio-sampling: engineers' discourse," *Popular Music*, vol. 10, no. 01, pp. 69–84, 1991.

[29] J. C. Brown, "Calculation of a constant Q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, p. 425, 1991.

[30] J. Haitsma, T. Kalker, and J. Oostveen, "Robust audio hashing for content identification," in *International Workshop on Content-Based Multimedia Indexing*, vol. 4, pp. 117–124, Citeseer, 2001.

[31] J. C. Brown, "An efficient algorithm for the calculation of a constant Q transform," *The Journal of the Acoustical Society of America*, vol. 92, no. 5, p. 2698, 1992.

[32] D. P. W. Ellis, "Spectrograms: Constant-Q (Log-frequency) and conventional (linear)," *http://www.ee.columbia.edu/˜dpwe/resources/matlab/sgram/.*

[33] M. Casey and M. Slaney, "Fast Recognition of Remixed Music Audio," in *Acoustics Speech and Signal Processing 2007 ICASSP 2007 IEEE International Conference on*, vol. 4, pp. 300–1, IEEE, 2007.

[34] M. Casey and M. Slaney, "Song Intersection by Approximate Nearest Neighbor Search," in *Proc. ISMIR*, pp. 144—-149, 2006.

[35] J. Serrà, M. Zanin, and R. G. Andrzejak, "Cover song retrieval by cross recurrence quantification and unsupervised set detection," in *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract; http://mtg.upf.edu/node/1517*, 2009.

[36] F. Kurth and M. Müller, "Efficient Index-Based Audio Matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 382–395, Feb. 2008.

[37] Shazam, "Company information, Who is Shazam?," *http://www.shazam.com/music/web/about.html*, 2011.

[38] D. Fragoulis, G. Rousopoulos, T. Panagopoulos, C. Alexiou, and C. Papaodysseus, "On the automated recognition of seriously distorted musical recordings," *Signal Processing, IEEE Transactions on*, vol. 49, no. 4, pp. 898–908, 2001.

[39] A. L.-C. Wang and J. O. Smith, "Landmark System and Methods for Recognizing Sound," *US Patent 6,990,453*, 2006.

[40] C. D. Manning, R. Prabhakar, and H. Schutze, *An introduction to Information Retrieval.* Cambridge University Press, 2008.

[41] M. Haro, "Detecting and Describing Percussive Events in Polyphonic Music," Master's thesis, 2008.

[42] R. C. Hendriks, R. Heusdens, and J. Jensen, "Perceptual linear predictive noise modelling for sinusoid-plus-noise audio coding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. IV, pp. 189–192, 2004.

[43] K. N. Hamdy, M. Ali, and A. H. Tewfi, "Low bit rate high quality audio coding with combined harmonic and wavelet representations," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 1045 –1048 vol. 2, 1996.

[44] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J Acoust Soc Am*, vol. 68, no. 5, p. 1523, 1980.

[45] H. Purwins and B. Blankertz, "CQ-profiles for key finding in audio," in *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract*, 2005.