Sound to Sense, Sense to Sound

A State of the Art in Sound and Music Computing

Pietro Polotti and Davide Rocchesso, editors

λογος

Pietro Polotti and Davide Rocchesso (Editors)
Dipartimento delle Arti e del Disegno Industriale
Università IUAV di Venezia
Dorsoduro 2206
30123 Venezia, Italia

This book has been developed within the Coordination Action $S2S^2$ (Sound to Sense, Sense to Sound) of the 6th Framework Programme - IST Future and Emerging Technologies: http://www.soundandmusiccomputing.org/

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

ISBN 978-3-8325-1600-0
Logos Verlag Berlin GmbH
Comeniushof, Gubener Str. 47,
10243 Berlin
Tel.: +49 (0)30 42 85 10 90
Fax: +49 (0)30 42 85 10 92
INTERNET: http://www.logos-verlag.de

Chapter 3

Content Processing of Music Audio Signals

Fabien Gouyon, Perfecto Herrera, Emilia Gómez, Pedro Cano, Jordi Bonada, Àlex Loscos, Xavier Amatriain, Xavier Serra

Music Technology Group, University Pompeu Fabra, Barcelona

About this chapter

In this chapter, we provide an overview of state-of-the-art algorithms for the automatic description of music audio signals, both from a low-level perspective (focusing on signal characteristics) and a more musical perspective (focusing on musically meaningful dimensions). We also provide examples of applications based on this description, such as music identification, music browsing and music signal transformations. Throughout the chapter, a special focus is put on promising research directions.

3.1 Introduction

Music Information Retrieval (MIR) is a young and very active research area. This is clearly shown in the constantly growing number and subjects of articles published in the Proceedings of the annual International Conference on Music Information Retrieval (ISMIR, the premier international scientific forum for researchers involved in MIR). MIR research is also increasingly published in high-standard scientific journals (e.g. the Communications of the ACM, or the IEEE Transactions on Audio, Speech and Language Processing)[1] and international conferences, as the ACM Multimedia, the International ACM SIGIR Conference, the IEEE International Conference on Acoustics, Speech, and Signal Processing, the IEEE International Conference on Multimedia and Expo, or the conference on Computer Music Modelling and Retrieval (CMMR), to name a few. In MIR, different long-established disciplines such as Musicology, Signal Processing, Psychoacoustics, Information Science, Computer Science, or Statistics converge by means of a multidisciplinary approach in order to address the wealth of scenarios for interacting with music posed by the digital technologies in the last decades (the standardisation of world-wide low-latency networks, the extensive use of efficient search engines in everyday life, the continuously growing amount of multimedia information on the web, in broadcast data streams or in personal and professional databases and the rapid development of on-line music stores). Applications are manifold; consider for instance automated music analysis, personalised music recommendation, online music access, query-based retrieval (e.g. "by-humming," "by-example") and automatic play-list generation. Among the vast number of disciplines and approaches to MIR (overviews of which can be found in Downie, 2003a and Orio, 2006), content processing of audio signals plays an important role. Music comes in many forms but content-based audio processing is only concerned with one of them: audio signals.[2] This chapter does not deal with the anal-

[1] both featuring recent special issues on MIR, see
`http://portal.acm.org/citation.cfm?id=1145287.1145308` and
`http://www.ewh.ieee.org/soc/sps/tap/sp_issue/cfp-mir.html`

[2] Hence the undifferentiated use in this chapter of the terms "music content processing" and "audio content processing."

ysis of symbolic music representations as e.g. digitised scores or structured representation of music events as MIDI. We also do not address the relatively new direction of research concerning the automated analysis of social, cultural and marketing dimensions of music networks (on these topics, we refer to Chapter 4, and the works by Cano et al., 2005b, 2006a and by Whitman, 2005). This section defines the notion of music content at diverse levels of abstraction and what we understand by *processing* music content: both its *description* and its *exploitation*. We also shortly mention representation issues in music content processing. Section 3.2 provides an overview of audio content description according to low-level features and diverse musically-meaningful dimensions as pitch, melody and harmony (see Section 3.2.4), rhythm (see Section 3.2.5), and music genre (see Section 3.2.6). The organisation follows increasing levels of abstraction. In Section 3.3, we address content exploitation and present different applications to content-based audio description. Finally, promising avenues for future work in the field are proposed in Section 3.4.

3.1.1 Music content: A functional view

A look at a dictionary reveals, at least, three senses for the word "content":

- Everything that is included in a collection;
- What a communication that is about something is about;
- The sum or range of what has been perceived, discovered or learned.

The disciplines of information science and linguistics offer interesting perspectives on the meaning of this term. However, we will rather focus on a more pragmatic view. The Society of Motion Picture and Television Engineers (SMPTE) and the European Broadcasting Union (EBU) have defined content as the combination of two entities termed *metadata* and *essence*. Essence is the raw program material itself, the data that directly encodes pictures, sounds, text, video, etc. Essence can also be referred to as media (although the former does not entail the physical carrier). In other words, essence is the encoded

information that directly represents the actual message, and it is normally presented in a sequential, time-dependent manner. On the other hand, metadata (literally, "data about the data") is used to *describe* the essence and its different manifestations. Metadata can be classified, according to SMPTE/EBU, into several categories:

- Essential (meta-information that is necessary to reproduce the essence, like the number of audio channels, the Unique Material Identifier, the video format, etc.);
- Access (to provide control and access to the essence, i.e. copyright information);
- Parametric (to define parameters of the essence capture methods like camera set-up, microphone set-up, perspective, etc.);
- Relational (to achieve synchronisation between different content components, e.g. time-code);
- Descriptive (giving a description of the actual content or subject matter in order to facilitate the cataloging, search, retrieval and administration of content; i.e. title, cast, keywords, classifications of the images, sounds and texts, etc.).

In a quite similar way the National Information Standards Organisation considers three main types of metadata:

- Descriptive metadata, which describe a resource for purposes such as discovery and identification; they can include elements such as title, abstract, author, and keywords.
- Structural metadata, which indicate how compound objects are put together, for example, how visual or audio takes are ordered to form a seamless audiovisual excerpt.
- Administrative metadata, which provide information to help manage a resource, such as "when" and "how" it was created, file type and other

technical information, and who can access it. There are several subsets of administrative data; two of them that are sometimes listed as separate metadata types are:

- Right Management metadata, which deals with intellectual property rights;
- Preservation metadata, which contains information needed to archive and preserve a resource.

In accordance with these rather general definitions of the term "metadata," we propose to consider as content *all that can be predicated from a media essence*. Any piece of information related to a music piece that can be annotated, extracted, and that is in any way meaningful (i.e. it carries semantic information) to some user, can be technically denoted as metadata. Along this rationale, the MPEG-7 standard defines a content descriptor as "a distinctive characteristic of the data which signifies something to somebody" (Manjunath et al., 2002). This rather permissive view on the nature of music contents has a drawback: as they represent many different aspects of a music piece, metadata are not certain to be understandable by *any* user. This is part of the "user-modelling problem," whose lack of precision participates in the so-called *semantic gap*, that is, "the lack of coincidence between the information that one can extract from the (sensory) data and the interpretation that the same data has for a user in a given situation" (Smeulders et al., 2000). That has been signaled by several authors (Smeulders et al., 2000; Lew et al., 2002; Jermyn et al., 2003) as one of the recurrent open issues in systems dealing with audiovisual content. It is therefore important to consider metadata together with their functional values and address the question of which content means what to which users, and in which application. A way to address this issue is to consider content hierarchies with different levels of abstraction, any of them potentially useful for *some* users. In that sense, think of how different a content description of a music piece would be if the targeted user was a naive listener or an expert musicologist. Even a low-level descriptor such as the spectral envelope of a signal can be thought of as a particular level of content description targeted for the signal processing engineer. All these specifically targeted descriptions can be thought of as different instantiations of the same general content description

scheme. Let us here propose the following distinction between descriptors of low, mid and high levels of abstraction (the latter being also sometimes referred to as "semantic" descriptors) (Lesaffre et al., 2003; Herrera, in print):

- A low-level descriptor can be computed from the essence data in a direct or derived way (i.e. after signal transformations like Fourier or Wavelet transforms, after statistical processing like averaging, after value quantisation like assignment of a discrete note name for a given series of pitch values, etc.). Most of low-level descriptors make little sense to the majority of users but, on the other hand, their exploitation by computing systems are usually easy. They can be also referred to as "signal-centered descriptors" (see Section 3.2.1).

- Mid-level descriptors require an induction operation that goes from available data towards an inferred generalisation about them. These descriptors usually pave the way for labelling contents, as for example a neural network model that makes decisions about music genre or about tonality, or a Hidden Markov Model that makes it possible to segment a song according to timbre similarities. Machine learning and statistical modelling make mid-level descriptors possible, but in order to take advantage of those techniques and grant the validity of the models, we need to gather large sets of observations. Mid-level descriptors are also sometimes referred to as "object-centered descriptors."

- The jump from low- or mid-level descriptors to high-level descriptors requires bridging the semantic gap. Semantic descriptors require an induction that has to be carried by means of a user-model (in order to yield the interpretation of the description), and not only a data-model as it was in the case of mid-level descriptors. As an example, let us imagine a simplistic "mood" descriptor consisting of labels "happy" and "sad." In order to compute such labels, one may[3] compute the tonality of the songs (i.e. "major" and "minor") and the tempo by means of knowledge-based analyzes of spectral and amplitude data. Using these mid-level descriptors, a model for computing the labels "happy" and "sad" would

[3] and it is only a speculation here

be elaborated by getting users' ratings of songs in terms of "happy" and "sad" and studying the relationships between these user-generated labels and values for tonality and tempo. High-level descriptors can also be referred to as "user-centered descriptors."

Standards In order to be properly exploited, music content (either low-, mid- or high-level content) has to be organised into knowledge structures such as taxonomies, description schemes, or ontologies. The Dublin Core and MPEG-7 are currently the most relevant standards for representing music content. The Dublin Core (DC) was specified by the Dublin Core Metadata Initiative, an institution that gathers organisations such as the Library of Congress, the National Science Foundation, or the Deutsche Bibliothek, to promote the widespread adoption of interoperable metadata standards. DC specifies a set of sixteen metadata elements, a core set of descriptive semantic definitions, which is deemed appropriate for the description of content in several industries, disciplines, and organisations. The elements are Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights, and Audience. Description, for example, can be an abstract, a table of contents, a graphical representation or free text. DC also specifies a list of qualifiers that refine the meaning and use of the metadata elements, which open the door to refined descriptions and controlled-term descriptions. DC descriptions can be represented using different syntaxes, such as HTML or RDF/XML. On the other hand, MPEG-7 is a standardisation initiative of the ISO/IEC Moving Picture Expert Group that, contrasting with other MPEG standards, does not address the *encoding* of audiovisual essence. MPEG-7 aims at specifying an interface for the description of multimedia contents. MPEG-7 defines a series of elements that can be used to describe content, but it does not specify the algorithms required to compute values for those descriptions. The building blocks of MPEG-7 description are descriptors, description schemes (complex structures made of aggregations of descriptors), and the Description Definition Language (DDL), which defines the syntax that an MPEG-7 compliant description has to follow. The DDL makes hence possible the creation of non-standard, but compatible, additional descriptors and description schemes. This is an important feature because different needs will

call for different kinds of structures, and for different instantiations of them. Depending on the theoretical and/or practical requirements of our problem, the required descriptors and description schemes will vary but, thanks to the DDL, we may build the proper structures to tailor our specific approach and required functionality. MPEG-7 descriptions are written in XML but a binary format has been defined to support their compression and streaming. The MPEG-7 standard definition covers eight different parts: Systems, DDL, Visual, Audio, Multimedia Description Schemes, Reference, Conformance and Extraction. In the audio section, we find music-specific descriptors for melody, rhythm or timbre, and in the Multimedia Description Schemes we find structures suitable to define classification schemes and a wealth of semantic information. As mentioned by Gómez et al. (2003a,b), the status of the original standard (see Manjunath et al., 2002, for an overview), as to representing music contents, is nevertheless a bit deceiving and it will probably require going beyond the current version for it to be adopted by the digital music community.

3.1.2 Processing music content: Description and exploitation

"Processing," beyond its straight meaning of "putting through a prescribed procedure," usually denotes a functional or computational approach to a wide range of scientific problems. "Signal processing" is the main term of reference here, but we could also mention "speech processing," "language processing," "visual processing" or "knowledge processing." A processing discipline focuses on the algorithmic level as defined by Marr (1982). The algorithmic level describes a system in terms of the steps that have to be carried out to solve a given problem. This type of description is, in principle, independent of the implementation level (as the algorithm can be effectively implemented in different ways). relationships and in terms of what is computed and why: for a given computational problem, several algorithms (each one implemented in several different ways) can be defined. The goal of a functional approach is that of developing systems that provide solutions to a given computational problem without considering the specific implementation of it. However, it is important to contrast the meaning of content processing with that of signal processing. The object of signal processing is the raw data captured by

sensors, whereas content processing deals with an object that is within the signal, embedded in it like a second-order code, and to which we refer to using the word metadata. The processes of extraction and modelling these metadata require the synergy of, at least, four disciplines: Signal Processing, Artificial Intelligence, Information Retrieval, and Cognitive Science. Indeed, they require, among other things:

- Powerful signal analysis techniques that make it possible to address complex real-world problems, and to exploit context- and content-specific constraints in order to maximise their efficacy.
- Reliable automatic learning techniques that help building models about classes of objects that share specific properties, about processes that show e.g. temporal trends.
- Availability of large databases of describable objects, and the technologies required to manage (index, query, retrieve, visualise) them.
- Usable models of the human information processing involved in the processes of extracting and exploiting metadata (i.e. how humans perceive, associate, categorise, remember, recall, and integrate into their behavior plans the information that might be available to them by means of other content processing systems).

Looking for the origins of music content processing, we can spot different forerunners depending on the contributing discipline that we consider. When focusing on the discipline of Information Retrieval, Kassler (1966) and Lincoln (1967) are among the acknowledged pioneers. The former defines music information retrieval as "the task of extracting, from a large quantity of music data, the portions of that data with respect to which some particular musicological statement is true" (p. 66) and presents a computer language for addressing those issues. The latter discusses three criteria that should be met for automatic indexing of music material: eliminating the transcription by hand, effective input language for music, and an economic means for printing the music. This thread was later followed by Byrd (1984), Downie (1994), McNab et al. (1996) and Blackburn (2000) with works dealing with score processing,

representation and matching of melodies as strings of symbols, or query by humming. Another batch of forerunners can be found when focusing on Digital Databases concepts and problems. Even though the oldest one dates back to the late eighties (Eaglestone, 1988), the trend towards databases for "content processing" emerges more clearly in the early nineties (de Koning and Oates, 1991; Eaglestone and Verschoor, 1991; Feiten et al., 1991; Keislar et al., 1995). These authors address the problems related to extracting and managing the acoustic information derived from a large amount of sound files. In this group of papers, we find questions about computing descriptors at different levels of abstraction, ways to query a content-based database using voice, text, and even external devices, and exploiting knowledge domain to enhance the functionalities of retrieval systems. To conclude with the antecedents for music content processing, we must also mention the efforts made since the last 30 years in the field of *Music Transcription*, whose goal is the automatic recovering of symbolic scores from acoustic signals (see Klapuri and Davy, 2006 for an exhaustive overview of music transcription research and Scheirer, 2000 for a critical perspective on music transcription). Central to music transcription is the segregation of the different music streams that coexist in a complex music rendition. Blind Source Separation (BSS) and Computational Auditory Scene Analysis (CASA) are two paradigms that address music stream segregation. An important conceptual difference between them is that, unlike the latter, the former intends to actually separate apart the different streams that summed together make up the multi-instrumental music signal. BSS is the agnostic approach to segregate music streams, as it usually does not assume any knowledge about the signals that have been mixed together. The strength of BSS models (but at the same time its main problem in music applications) is that only mutual statistical independence between the source signals is assumed, and no a priori information about the characteristics of the source signals (Casey and Westner, 2000; Smaragdis, 2001). CASA, on the other hand, is partially guided by the groundbreaking work of Bregman (1990) – who originally coined the term "Auditory Scene Analysis" (ASA) – on the perceptual mechanisms that enables a human listener to fuse or fission concurrent auditory events. CASA addresses the computational counterparts of ASA. Computer systems embedding ASA theories assume, and implement, specific

heuristics that are hypothesised to play a role in the way humans perceive the music, as for example Gestalt principles. Worth mentioning here are the works by Mellinger (1991), Brown (1992), Ellis (1996), Kashino and Murase (1997), and Wang and Brown (2006). A comprehensive characterisation of the field of music content processing was offered by Leman (2003): "the science of musical content processing aims at explaining and modelling the mechanisms that transform information streams into meaningful musical units (both cognitive and emotional)." Music content processing is, for Leman, the object of study of his particular view of musicology, much akin to the so-called systematic musicology than to historic musicology. He additionally provides a definition of music content processing by extending it along three dimensions:

- The intuitive-speculative dimension, which includes semiotics of music, musicology, sociology, and philosophy of music. These disciplines provide a series of concepts and questions from a culture-centric point of view; music content is, following this dimension, a culture-dependent phenomenon.

- The empirical-experimental dimension, which includes research in physiology, psychoacoustics, music psychology, and neuro-musicology. These disciplines provide most of the empirical data needed to test, develop or ground some elements from the intuitive dimension; music content is, following this dimension, a percept in our auditory system.

- The computation-modelling dimension, which includes sound analysis and also computational modelling and simulation of perception, cognition and action. Music content is, following this dimension, a series of processes implemented in a computer, intended to emulate a human knowledge structure.

One can argue that these three dimensions address only the descriptive aspect of music content processing. According to Aigrain (1999), "content processing is meant as a general term covering feature extraction and modelling techniques for enabling basic retrieval, interaction and creation functionality." He also argues that music content processing technologies will provide "new

aspects of listening, interacting with music, finding and comparing music, performing it, editing it, exchanging music with others or selling it, teaching it, analyzing it and criticizing it." We see here that music content processing can be characterised by two different tasks: *describing* and *exploiting* content. Furthermore, as mentioned above, the very meaning of "music content" cannot be entirely grasped without considering its functional aspects and including specific applications, targeted to specific users (Gouyon and Meudic, 2003). Hence, in addition to describing music content (as reviewed in Section 3.2), music content processing is also concerned with the design of computer systems that open the way to a more pragmatic content exploitation according to constraints posed by Leman's intuitive, empirical and computational dimensions (this exploitation aspect is the subject of Section 3.3).

3.2 Audio content description

3.2.1 Low-level audio features

Many different low-level features can be computed from audio signals. Literature in signal processing and speech processing provides us with a dramatic amount of techniques for signal modelling and signal representations over which features can be computed. Parametric methods (e.g. AR modelling, Prony modelling) directly provide such features, while additional post-processing is necessary to derive features from non-parametric methods (e.g. peaks can be extracted from spectral or cepstral representations). A comprehensive overview of signal representation and modelling techniques and their associated features is clearly beyond the scope of this chapter. Thus, we will only mention some features commonly used in music audio signal description, with a special focus on work published in the music transcription and MIR literature. Commonly, the audio signal is first digitised (if necessary) and converted to a general format, e.g. mono PCM (16 bits) with a fixed sampling rate (ranging from 5 to 44.1 KHz). A key assumption is that the signal can be regarded as being stationary over intervals of a few milliseconds. Therefore, the signal is divided into frames (short chunks of signal) of for example 10 ms. The number

of frames computed per second is called *frame rate*. A tapered window function (e.g. a Gaussian or Hanning window) is applied to each frame to minimise the discontinuities at the beginning and end. Consecutive frames are usually considered with some *overlap* for smoother analyzes. The analysis step, the *hop size*, equals the frame rate minus the overlap.

Temporal features Many audio features can be computed directly from the temporal representation of these frames, for instance, the *mean* (but also the *maximum* or the *range*) of the amplitude of the samples in a frame, the *energy*, the *zero-crossing rate*, the *temporal centroid* (Gómez et al., 2005) and *auto-correlation coefficients* (Peeters, 2004). Some low-level features have also shown to correlate with perceptual attributes, for instance, amplitude is loosely correlated with *loudness*.

Spectral features It is also very common to compute features on a different representation of the audio, as for instance the spectral representation. Hence, a spectrum is obtained from each signal frame by applying a Discrete Fourier Transform (DFT), usually with the help of the Fast Fourier Transform (FFT). This procedure is called Short-Time Fourier Transform (STFT). Sometimes, the time-frequency representation is further processed by taking into account perceptual processing that takes place in the human auditory system as for instance the filtering performed by the middle-ear, loudness perception, temporal integration or frequency masking (Moore, 1995). Many features can be computed on the obtained representation, e.g. the spectrum *energy*, energy values in several frequency sub-bands (e.g. the perceptually-motivated *Bark bands*, Moore, 1995), the *mean*, *geometric mean*, *spread*, *centroid*, *flatness*, *kurtosis*, *skewness*, *spectral slope*, *high-frequency content* and *roll-off* of the spectrum frequency distribution or the *kurtosis* and *skewness* of the spectrum magnitude distribution (see Peeters, 2004 and Gómez et al., 2005 for more details on these numerous features). Further modelling of the spectral representation can be achieved through sinusoidal modelling (McAulay and Quatieri, 1986) or sinusoidal plus residual modelling (Serra, 1989). Other features can be computed on the series of *spectral peaks* corresponding to each frame and on the spectrum

of the *residual* component. Let us mention, for instance, the mean (and the accumulated) *amplitude* of sinusoidal and residual components, the *noisiness*, the *harmonic distortion*, the *harmonic spectral centroid*, the *harmonic spectral tilt* and different ratios of peak amplitudes as the first, second and third *tristimulus* or the *odd-to-even ratio* (Serra and Bonada, 1998; Gómez et al., 2005). Bear in mind that other transforms can be applied instead of the DFT such as the Wavelet (Kronland-Martinet et al., 1987) or the Wigner-Ville transforms (Cohen, 1989).

Cepstral features *Mel-Frequency Cepstrum Coefficients* (MFCCs) are widespread descriptors in speech research. The cepstral representation has been shown to be of prime importance in this field, partly because of its ability to nicely separate the representation of voice excitation (the higher coefficients) from the subsequent filtering performed by the vocal tract (the lower coefficients). Roughly, lower coefficients represent spectral envelope (i.e. the formants) while higher ones represent finer details of the spectrum, among them the pitch (Oppenheim and Schafer, 2004). One way of computing the Mel-Frequency Cepstrum from a magnitude spectrum is the following:

1. Projection of the frequency axis from linear scale to the Mel scale of lower dimensionality (i.e. 20, by summing magnitudes in each of the 20 frequency bands of a Mel critical-band filter-bank);
2. Magnitude logarithm computation;
3. Discrete Cosine Transform (DCT).

The number of output coefficients of the DCT is variable. It is often set to 13, as in the standard implementation of the MFCCs detailed in the widely-used speech processing software Hidden Markov Model Toolkit (HTK).[4]

Temporal evolution of frame features Apart from the instantaneous, or frame, feature values, many authors focus on the temporal evolution of features (see Meng, 2006, for an overview). The simplest way to address temporal

[4] `http://htk.eng.cam.ac.uk/`

evolution of features is to compute the derivative of feature values (which can be estimated by a first-order differentiator). The degree of change can also be measured as the feature differential normalised with its magnitude (Klapuri et al., 2006). This is supposed to provide a better emulation of human audition, indeed, according to Weber's law, for humans, the just-noticeable-difference in the increment of a physical attribute depends linearly on its magnitude before incrementing. That is, $\Delta x/x$ (where x is a specific feature and Δx is the smallest perceptual increment) would be constant.

3.2.2 Segmentation and region features

Frame features represent a significant reduction of dimensionality with respect to the audio signal itself, however, it is possible to further reduce the dimensionality by focusing on features computed on groups of consecutive frames (Meng, 2006), often called *regions*. An important issue here is the determination of relevant region boundaries: i.e. the *segmentation* process. Once a given sound has been segmented into regions, it is possible to compute features as statistics of all of the frame features over the whole region (Serra and Bonada, 1998).

Segmentation Segmentation comes in different flavors. For McAdams and Bigand (1993), it "refers to the process of dividing an event sequence into distinct groups of sounds. The factors that play a role in segmentation are similar to the grouping principles addressed by Gestalt psychology." This definition implies that the segmentation process represents a step forward in the level of abstraction of data description. However, it may not necessarily be the case. Indeed, consider an adaptation of a classic definition coming from the visual segmentation area (Pal and Pal, 1993): "[sound] segmentation is a process of partitioning [the sound file/stream] into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous." The notion of homogeneity in this definition implies a property of signal or feature stationarity that may equate to a perceptual grouping process, but not necessarily. In what is sometimes referred to as *model-free*

segmentation, the main idea is using the amount of change of a feature vector as a boundary detector: when this amount is higher than a given threshold, a boundary change decision is taken. Threshold adjustment requires a certain amount of trial-and-error, or fine-tuned adjustments regarding different segmentation classes. Usually, a smoothing window is considered in order to weight contributions from closer observations (Vidal and Marzal, 1990, p. 45). It is also possible to generalise the previous segmentation process to multi-dimensional feature vectors. There, the distance between consecutive frames can be computed with the help of different measures as for example the Mahalanobis distance (Tzanetakis and Cook, 1999). In the same vein, Foote (2000) uses MFCCs and the cosine distance measure between pairs of frames (not only consecutive frames), which yields a dissimilarity matrix that is further correlated with a specific kernel. Different kernels can be used for different types of segmentations (from short- to long-scale). The level of abstraction that can be attributed to the resulting regions may depend on the features used in the first place. For instance, if a set of low-level features is known to correlate strongly with a human percept (as the fundamental frequency correlates with the pitch and the energy in Bark bands correlates with the loudness) then the obtained regions may have some relevance as features of mid-level of abstraction (e.g. music notes in this case). *Model-based* segmentation on the other hand is more directly linked to the detection of mid-level feature boundaries. It corresponds to a focus on mid-level features that are thought, *a priori*, to make up the signal. A classical example can be found in speech processing where dynamical models of phonemes, or words, are built from observations of labelled data. The most popular models are Hidden Markov Models (HMM) (Rabiner, 1989). Applications of HMMs to the segmentation of music comprise segmentation of fundamental frequency envelopes in music notes (Raphael, 1999) and segmentation of MFCC-based temporal series in regions of globally-homogeneous timbres (Batlle and Cano, 2000). Rossignol (2000) proposes other examples of model-based segmentation and reports on the performance of different induction algorithms – Gaussian Mixture Models (GMM), k-Nearest Neigbours (k-NN) and Artificial Neural Networks (ANN) – in the tasks of speech/music segmentation and intra-note segmentation (see also Chapter 4). In the more general context of signal segmentation (not just

music signals), Basseville and Nikiforov (1993) propose many segmentation techniques, some of which entail the use of signal models. For instance, they propose a time-domain technique in which two temporal windows are used: a sliding window with fixed size and a window with constant size increase. In this technique, a distance estimation is computed between two AR models built on each window (derived from the cross entropy between the conditional distributions of the two models). Here also, a threshold is used to determine whether the distance should be considered representative of a boundary or not. Application of this technique to music signals can be found in the works by Jehan (1997) and Thornburg and Gouyon (2000).

Note onset detection The detection of note onsets in music signals has attracted many computer music researchers since the early eighties (Gordon, 1984). Several methods have been designed, making use of diverse low-level features. The simplest focus on the temporal variation of a single feature, for instance the energy or the pitch. However, the combined use of multiple features (as energy *and* pitch) seems to provide better estimates, state-of-the-art algorithms often making use of band-wise energy processing (Klapuri, 1999; Bello, 2003). Model-based note onset segmentation has also been an active research field (Thornburg and Gouyon, 2000). The literature on onset detection is extensive and a review is beyond the scope of this chapter (for an exhaustive overview, see Bello, 2003; Bello et al., 2005).

Intra-note segmentation In addition to note onset detection, some research has also been dedicated to the segmentation of music signals in terms of Attack, Sustain and Release regions. This is especially relevant, from a feasibility point of view, when dealing with isolated instrument samples or musical phrases played by a monophonic instrument (Jenssen, 1999; Maestre and Gómez, 2005). Given starting and ending boundaries of these regions, it is possible to compute a number of features that relate to their durations as for example the *log-attack time* (Peeters, 2004). Some authors also focus on the variations of low-level frame features in these regions, such as the energy (Maestre and Gómez, 2005) or the fundamental frequency in sustain regions, characterizing therefore the

vibrato (Herrera and Bonada, 1998; Rossignol et al., 1999; Collins, 2005).

Speech/Music segmentation A large body of work in automatic segmentation of audio signal also concerns the determination of boundaries of speech regions and music regions. This is usually achieved by model-based segmentation of multiple low-level features (Scheirer and Slaney, 1997; Harb and Chen, 2003; Pinquier et al., 2003; Kotti et al., 2006).

3.2.3 Audio fingerprints

Audio fingerprints have attracted a lot of attention for their usefulness in audio identification applications (see Section 3.3). Audio fingerprints are compact content-based signatures summarizing audio recordings (e.g. energies in specific frequency bands) that can be extracted from a music audio piece and stored in a database. Fingerprints of unlabelled pieces of audio can be calculated and matched against those stored in the database, providing a link to corresponding metadata (e.g. artist and song name). Section 3.3 provides more details on the main requirements of fingerprinting systems and application scenarios (for a general functional framework of audio fingerprinting systems and an overview of current technologies, see Cano et al., 2005a).[5] This section provides a short overview of audio features commonly used in the design of audio fingerprints.

Fingerprint extraction The fingerprint extraction derives a set of features from a recording in a concise and robust form. Fingerprint requirements include:

- Discrimination power over huge numbers of other fingerprints;
- Invariance to distortions;
- Compactness;

[5]Note that "fingerprinting" should not be mistaken for "watermarking," differences are explained in (Gomes et al., 2003).

- Computational simplicity.

The simplest approach one may think of – using directly the digitised waveform – is neither efficient nor effective. A more efficient implementation of this approach could use a hash method such as MD5 (Message Digest 5) or CRC (Cyclic Redundancy Checking) to obtain a compact representation of the binary file. However, hash values are fragile, a single bit flip is sufficient for the hash to completely change. They are also not robust to compression or distortions. Most fingerprint extraction systems consist of a front-end and a fingerprint modelling block (see Figure 3.1). The front-end computes low-level features from the signal and the fingerprint model defines the final fingerprint representation; we now briefly describe them in turn.

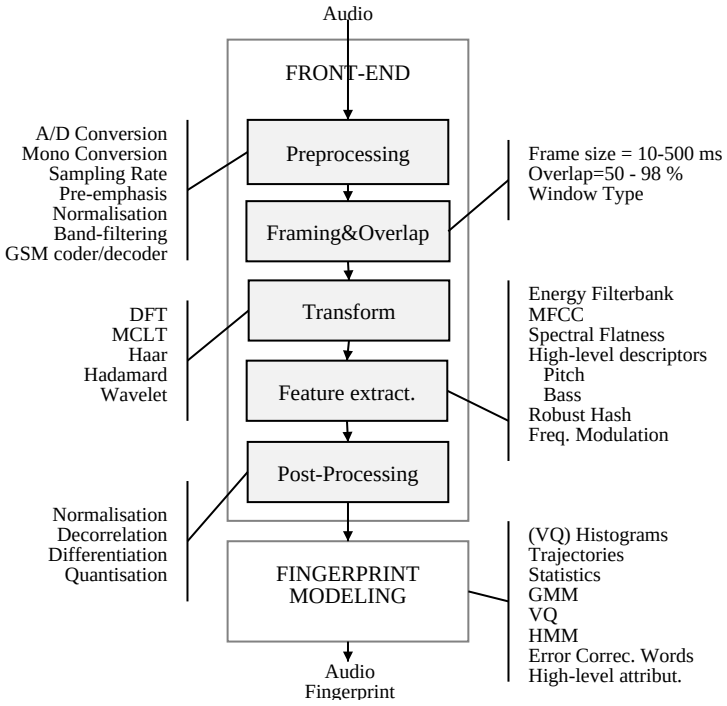

Figure 3.1: Fingerprint Extraction Framework: Front-end (top) and Fingerprint modelling (bottom).

Front-end Several driving forces co-exist in the design of the front-end: dimensionality reduction, perceptually meaningful parameters (similar to those used by the human auditory system), invariance/robustness (to channel distortions, background noise, etc.) and temporal correlation (systems that capture spectral dynamics). After the first step of audio digitisation, the audio is sometimes preprocessed to simulate the channel, for example band-pass filtered in a telephone identification task. Other types of processing are a GSM coder/decoder in a mobile phone identification system, pre-emphasis, amplitude normalisation (bounding the dynamic range to $[-1, 1]$). After framing the signal in small windows, overlap must be applied to assure robustness to shifting (i.e. when the input data is not perfectly aligned to the recording that was used for generating the fingerprint). There is a trade-off between the robustness to shifting and the computational complexity of the system: the higher the frame rate, the more robust to shifting the system is but at a cost of a higher computational load. Then, linear transforms are usually applied (see Figure 3.1). If the transform is suitably chosen, the redundancy is significantly reduced. There are optimal transforms in the sense of information packing and de-correlation properties, like Karhunen-Loeve (KL) or Singular Value Decomposition (SVD). These transforms, however, are computationally complex. For that reason, lower complexity transforms using fixed basis vectors are common (e.g. the DFT). Additional transformations are then applied in order to generate the final acoustic vectors. In this step, we find a great diversity of algorithms. The objective is again to reduce the dimensionality and, at the same time, to increase the invariance to distortions. It is very common to include knowledge of the transduction stages of the human auditory system to extract more perceptually meaningful parameters. Therefore, many systems extract several features performing a critical-band analysis of the spectrum. Resulting features are for example MFCCs, energies in Bark-scaled bands, geometric mean of the modulation frequency, estimation of the energy in Bark-spaced band-filters, etc., or many of the features presented in Section 3.2.1. Some examples are given in Figure 3.2. Most of the features described so far are absolute measurements. In order to better characterise temporal variations in the signal, higher order time derivatives are added to the signal model. Some systems compact the feature vector representation

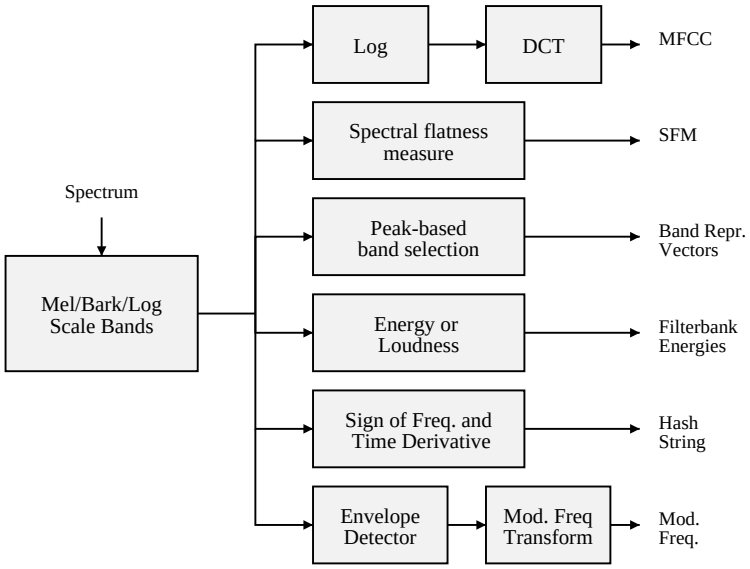

Figure 3.2: Feature Extraction Examples

using transforms such as Principal Component Analysis (PCA). It is also quite common to apply a very low resolution quantisation (ternary or binary) to the features, the purpose of which is to gain robustness against distortions and reduce the memory requirements.

Fingerprint models The sequence of features calculated on a frame-by-frame basis is then further reduced to a fingerprint model that usually implies statistics of frame values (mean and variance) and redundancies in frame vicinity. A compact representation can also be generated by clustering the feature vectors. The sequence of vectors is thus approximated by a much lower number of representative code vectors, a codebook. The temporal evolution of audio is lost with this approximation, but can be kept by collecting short-time statistics over regions of time or by HMM modelling (Batlle et al., 2002). At that point, some systems also derive musically-meaningful attributes from low-level features, as the *beats* (Kirovski and Attias, 2002) (see Section 3.2.5) or the *predominant pitch* (Blum et al., 1999) (see Section 3.2.4).

3.2.4 Tonal descriptors: From pitch to key

This section first reviews computational models of *pitch* description and then progressively addresses tonal aspects of higher levels of abstraction that imply different combinations of pitches: *melody* (sequence of single pitches combined over time), *pitch classes* and *chords* (simultaneous combinations of pitches), and *chord progressions*, *harmony* and *key* (temporal combinations of chords).

Pitch

The fundamental frequency is the main low-level descriptor to consider when describing melody and harmony. Due to the significance of pitch detection for speech and music analysis, a lot of research has been done in this field. We present here a brief review of the different approaches for pitch detection: fundamental frequency estimation for monophonic sounds, multi-pitch estimation and predominant pitch estimation. We refer to the paper by Gómez et al. (2003c) for an exhaustive review.

Fundamental frequency estimation for monophonic sounds As illustrated in Figure 3.3, the fundamental frequency detection process can be subdivided into three successive steps: the preprocessor, the basic extractor, and the post-processor (Hess, 1983). The basic extractor converts the input signal into a series of fundamental frequency estimates, one per analysis frame. Pitched/unpitched measures are often additionally computed to decide whether estimates are valid or should be discarded (Cano, 1998). The main task of the preprocessor is to facilitate the fundamental frequency extraction. Finally, the post-processor performs more diverse tasks, such as error detection and correction, or smoothing of an obtained contour. We now describe these three processing blocks in turn. Concerning the main extractor processing block, the first solution was to adapt the techniques proposed for speech (Hess, 1983). Later, other methods have been specifically designed for dealing with music signals. These methods can be classified according to their processing domain: time-domain algorithms vs frequency-domain algorithms. This distinction is not always so clear, as some of the algorithms can be expressed in both (time

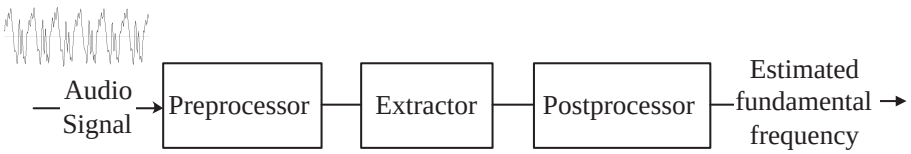

Figure 3.3: Steps of the fundamental frequency detection process

and frequency) domains, as the autocorrelation function (ACF) method. Another way of classifying the different methods, more adapted to the frequency domain, is to distinguish between spectral place algorithms and spectral interval algorithms (Klapuri, 2004). The spectral place algorithms weight spectral components according to their spectral location. Other systems use the information corresponding to spectral intervals between components. Then, the spectrum can be arbitrarily shifted without affecting the output value. These algorithms work relatively well for sounds that exhibit inharmonicity, because intervals between harmonics remain more stable than the places for the partials.

Time-domain algorithms The simplest time-domain technique is based on counting the number of times the signal crosses the 0-level reference, the zero-crossing rate (ZCR). This method is not very accurate when dealing with noisy signals or harmonic signals where the partials are stronger than the fundamental. Algorithms based on the time-domain autocorrelation function (ACF) have been among the most frequently used fundamental frequency estimators. ACF-based fundamental frequency detectors have been reported to be relatively noise immune but sensitive to formants and spectral peculiarities of the analyzed sound (Klapuri, 2004). Envelope periodicity algorithms find their roots in the observation that signals with more than one frequency component exhibit periodic fluctuations in their time domain amplitude envelope. The rate of these fluctuations depends on the frequency difference between the two frequency components. In the case of a harmonic sound, the fundamental frequency is clearly visible in the amplitude envelope of the signal. Recent models of human pitch perception tend to calculate envelope periodicity separately in distinct frequency bands and then combine the results

across channels (Meddis and Hewitt, 1991; Terhardt et al., 1981). These methods attempt to estimate the perceived pitch, not only the physical periodicity, in acoustic signals of various kinds. A parallel processing approach (Gold and Rabiner, 1969; Rabiner and Schafer, 1978), designed to deal with speech signals, has been successfully used in a wide variety of applications. Instead of designing one very complex algorithm, the basic idea is to tackle the same problem with several, more simple processes in parallel and later combine their outputs. As Bregman (1998) points out, human perception appears to be redundant at many levels, several different processing principles seem to serve the same purpose, and when one of them fails, another is likely to succeed.

Frequency-domain algorithms another transformation. Noll (1967) introduced the idea of Cepstrum analysis for pitch determination of speech signals. The cepstrum is the inverse Fourier transform of the power spectrum logarithm of the signal. The Cepstrum computation (see Section 3.2.1) nicely separates the transfer function (spectral envelope) from the source, hence the pitch. Cepstrum fundamental frequency detection is closely similar to autocorrelation systems (Klapuri, 2004). Spectrum autocorrelation methods were inspired by the observation that a periodic but non-sinusoidal signal has a periodic magnitude spectrum, the period of which is the fundamental frequency. This period can be estimated by ACF (Klapuri, 2004). Harmonic matching methods extract a period from a set of spectral peaks of the magnitude spectrum of the signal. Once these peaks in the spectrum are identified, they are compared to the predicted harmonics for each of the possible candidate note frequencies, and a fitness measure can be developed. A particular fitness measure is described by Maher and Beauchamp (1993) as a "Two Way Mismatch" procedure. This method is used in the context of Spectral Modelling Synthesis (SMS), with some improvements, as pitch-dependent analysis window, enhanced peak selection, and optimisation of the search (Cano, 1998).

The idea behind Wavelet-based algorithms is to filter the signal using a wavelet with derivative properties. The output of this filter will have maxima where glottal-closure instants or zero crossings happen in the input signal. After detection of these maxima, the fundamental frequency can be estimated

as the distance between consecutive maxima.

Klapuri (2004) proposes a band-wise processing algorithm that calculates independent fundamental frequency estimates in separate frequency bands. Then, these values are combined to yield a global estimate. This method presents several advantages: it solves the inharmonicity problem, it is robust with respect to heavy signal distortions, where only a fragment of the frequency range is reliable.

Preprocessing methods The main task of a preprocessor is to suppress noise prior to fundamental frequency estimation. Some preprocessing methods used in speech processing are detailed in the book by Hess (1983). Methods specifically defined for music signals are detailed in the already mentioned work by Klapuri (2004).

Post-processing methods The estimated series of pitches may be noisy and may present isolated errors, different methods have been proposed for correcting these. The first is low-pass filtering (linear smoothing) of the series. This may remove much of the local jitter and noise, but does not remove local gross measurement errors, and, in addition, it smears the intended discontinuities at the voiced-unvoiced transitions (Hess, 1983). Non-linear smoothing has been proposed to address these problems (Rabiner et al., 1975). Another procedure consists in storing several possible values for the fundamental frequency for each analysis frame (Laroche, 1995), assigning them a score (e.g. the value of the normalised autocorrelation). Several tracks are then considered and ranked (according to some continuity evaluation function) by for example dynamic programming. This approach minimises the abrupt fundamental frequency changes (e.g. octave errors) and gives good results in general. Its main disadvantage is its estimation delay and non-causal behavior. Usually, it is also useful to complement the forward estimation by a backward estimation (Cano, 1998).

Multi-pitch estimation Multi-pitch estimation is the simultaneous estimation of the pitches making up a polyphonic sound (a polyphonic instrument or

several instruments playing together). Some algorithms used for monophonic pitch detection can be adapted to the simplest polyphonic situations (Maher and Beauchamp, 1993). However, they are usually not directly applicable to general cases, they require, among other differences, significantly longer time frames (around 100 ms) (Klapuri, 2004). Relatively successful algorithms implement principles of the perceptual mechanisms that enable a human listener to fuse or fission concurrent auditory streams (see references to "Auditory Scene Analysis" on page 92). For instance, Kashino et al. (1995) implement such principles in a Bayesian probability network, where bottom-up signal analysis can be integrated with temporal and musical predictions. An example following the same principles is detailed by Walmsley et al. (1999), where a comparable network estimates the parameters of a harmonic model jointly for a number of frames. Godsmark and Brown (1999) have developed a model that is able to resolve melodic lines from polyphonic music through the integration of diverse knowledge. Other methods are listed in the work by Klapuri (2004). The state-of-the-art multi-pitch estimators operate reasonably well for clean signals, frame-level error rates increasing progressively with the number of concurrent voices. Also, the number of concurrent voices is often underestimated and the performance usually decreases significantly in the presence of noise (Klapuri, 2004).

Predominant pitch estimation Predominant pitch estimation also aims at estimating pitches in polyphonic mixtures; however, contrarily to multi-pitch estimation, it assumes that a specific instrument is predominant and defines the melody. For instance, the system proposed by Goto (2000) detects melody and bass lines in polyphonic recordings using a multi-agent architecture by assuming that they occupy different frequency regions. Other relevant methods are reviewed by Gómez et al. (2003c) and Klapuri (2004).

Melody

Extracting melody from note sequences We have presented above several algorithms whose outputs are time sequences of pitches (or simultaneous

combinations thereof). Now, we present some approaches that, building upon those, aim at identifying the notes that are likely to correspond to the main melody. We refer to the paper by Gómez et al. (2003c) for an exhaustive review of the state-of-the-art in melodic description and transformation from audio recordings. Melody extraction can be considered not only for polyphonic sounds, but also for monophonic sounds as they may contain notes that do not belong to the melody (for example, grace notes, passing notes or the case of several interleaved voices in a monophonic stream). As discussed by Nettheim (1992) and Selfridge-Field (1998, Section 1.1.3.), the derivation of a melody from a sequence of pitches faces the following issues:

- A single line played by a single instrument or voice may be formed by movement between two or more melodic or accompaniment strands.
- Two or more contrapuntal lines may have equal claim as "the melody."
- The melodic line may move from one voice to another, possibly with overlap.
- There may be passages of figuration not properly considered as melody.

Some approaches try to detect note groupings. Experiments have been done on the way listeners achieve melodic grouping (McAdams, 1994; Scheirer, 2000, p.131). These provide heuristics that can be taken as hypotheses in computational models. Other approaches make assumptions on the type of music to be analyzed. For instance, methods can be different according to the complexity of the music (monophonic or polyphonic music), the genre (classical with melodic ornamentations, jazz with singing voice, etc.) or the representation of the music (audio, MIDI, etc.). We refer to the works by Uitdenbogerd and Zobel (1998) and by Typke (2007) regarding melody extraction of MIDI data.

Melodic segmentation The goal of melodic segmentation is to establish a temporal structure on a sequence of notes. It may involve different levels of hierarchy, such as those defined by Lerdahl and Jackendoff (1983), and may include overlapping, as well as unclassified, segments. One relevant method

proposed by Cambouropoulos (2001) is the Local Boundary Detection Model (LBDM). This model computes the transition strength of each interval of a melodic surface according to local discontinuities. function are considered as segment boundaries. This method is based on two rules: the *Change Rule* (measuring the degree of change between two consecutive intervals) and the *Proximity Rule* (each boundary is weighted according to the size of its absolute interval, so that segment boundaries are located at larger intervals). In his paper, Cambouropoulos (2001) uses not only pitch, but also temporal (inter-onset intervals, IOIs) and rest intervals. He compares this algorithm with the punctuation rules defined by Friberg and colleagues,[6] getting coherent results. The LBDM has been used by Melucci and Orio (1999) for content-based retrieval of melodies. Another approach can be found in the `Grouper`[7] module of the `Melisma` music analyzer, implemented by Temperley and Sleator. This module uses three criteria to select the note boundaries. The first one considers the gap score for each pair of notes that is the sum of the IOIs and the offset-to-onset interval (OOI). Phrases receive a weight proportional to the gap score between the notes at the boundary. The second one considers an optimal phrase length in number of notes. The third one is related to the metrical position of the phrase beginning, relative to the metrical position of the previous phrase beginning. Spevak et al. (2002) have compared several algorithms for melodic segmentation: LBDM, the Melisma Grouper, and a memory-based approach, the Data-Oriented Parsing (DOP) by Bod (2001). They also describe other approaches to melodic segmentation. To explore this issue, they have compared manual segmentation of different melodic excerpts. However, according to them, "it is typically not possible to determine one 'correct' segmentation, because the process is influenced by a rich and varied set of context."

Miscellaneous melodic descriptors Other descriptors can be derived from a numerical analysis of the pitches of a melody and used in diverse applications as comparative analysis (Toiviainen and Eerola, 2001), melody re-

[6] see `http://www.speech.kth.se/music/performance/performance_rules.html`, see also Chapter 7

[7] see `http://www.link.cs.cmu.edu/music-analysis/grouper.html`

trieval (Kostek, 1998; Tzanetakis, 2002), and algorithmic composition (Towsey et al., 2001). Some of these descriptors are computed using features related to structural, musical or perceptual aspects of sound. Some others are computed from note descriptors (therefore they require algorithms for note segmentation, see Section 3.2.2). Yet other descriptors can be computed as statistics of frame or sample features. One example is the pitch histogram features proposed by Tzanetakis (2002).

Pitch class distribution

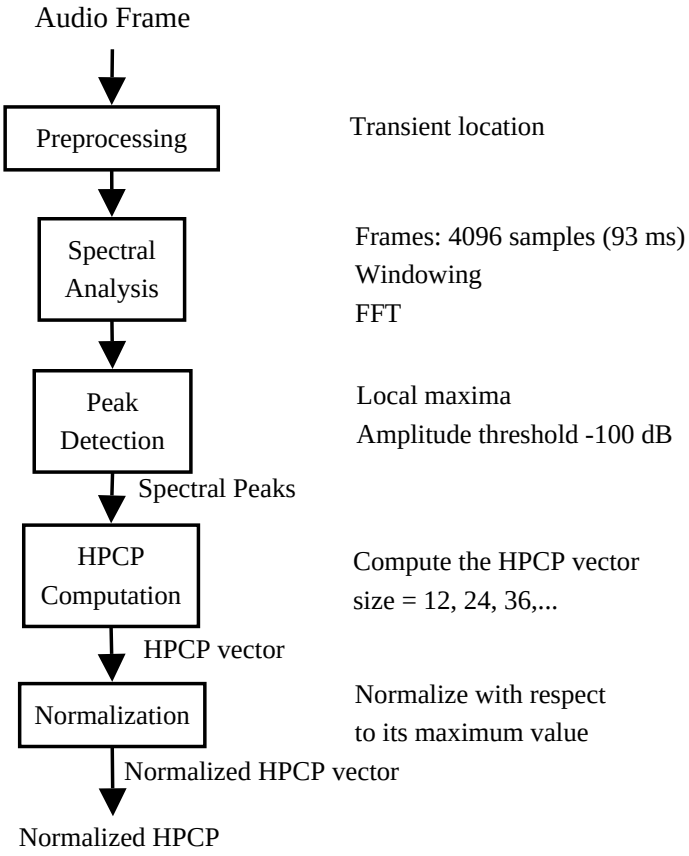

Figure 3.4: Block Diagram for HPCP Computation

Many efforts have been devoted to the analysis of chord sequences and key in MIDI representations of classical music, but little work has dealt di-

rectly with audio signals and other music genres. Adapting MIDI-oriented methods would require a previous step of automatic transcription of polyphonic audio, which, as mentioned by Scheirer (2000) and Klapuri (2004), is far from being solved. Some approaches extract information related to the pitch class distribution of music without performing automatic transcription. The pitch-class distribution is directly related to the chords and the tonality of a piece. Chords can be recognised from the pitch class distribution without requiring the detection of individual notes. Tonality can be also estimated from the pitch class distribution without a previous procedure of chord estimation. Fujishima (1999) proposes a system for chord recognition based on the pitch-class profile (PCP), a 12-dimensional low-level vector representing the intensities of the twelve semitone pitch classes. His chord recognition system compares this vector with a set of chord-type templates to estimate the played chord. In a paper by Sheh and Ellis (2003), chords are estimated from an audio recordings by modelling sequences of PCPs with an HMM. In the context of a key estimation system, Gómez (2004) proposes the Harmonic PCPs (HPCPs) as extension of the PCPs: only the spectral peaks in a certain frequency band are used ($100 - 5000$ Hz), a weight is introduced into the feature computation and a higher resolution is used in the HPCP bins (decreasing the quantisation level to less than a semitone). The procedure for HPCP computation is illustrated in Figure 3.4. A transient detection algorithm (Bonada, 2000) is used as preprocessing step in order to discard regions where the harmonic structure is noisy; the areas located 50 ms before and after the transients are not analyzed. As a post-processing step, HPCPs are normalised with respect to maximum values for each analysis frame, in order to store the relative relevance of each of the HPCP bins. In the context of beat estimation of drum-less audio signals, Goto and Muraoka (1999) also introduced the computation of a histogram of frequency components, used to detect chord changes. Note however that this method does not identify chord names. Constant Q profiles have also been used to characterise the tonal content of audio (Purwins et al., 2000). Constant Q profiles are twelve-dimensional vectors, each component referring to a pitch class, which are computed with the constant Q filter bank (Brown and Puckette, 1992). Purwins et al. (2003) present examples where constant Q profiles are used to track tonal centers. Later on, Purwins (2005) uses these features to

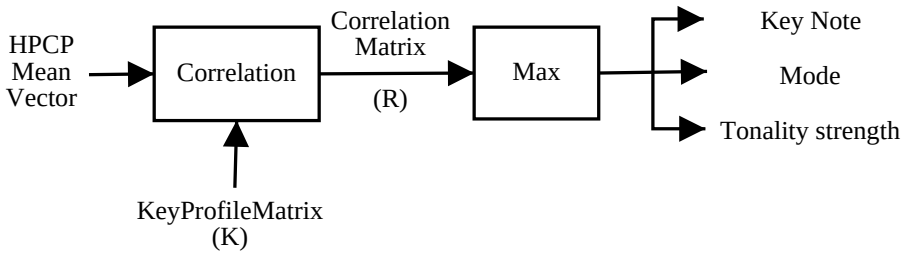

Figure 3.5: Block Diagram for Key Computation using HPCP

analyze the interdependence of pitch classes and key as well as key and composer. Tzanetakis (2002) proposes a set of features related to audio harmonic content in the context of music genre classification. These features are derived from a pitch histogram that can be computed from MIDI or audio data: the most common pitch class used in the piece, the frequency of occurrence of the main pitch class and the pitch range of a song.

Tonality: From chord to key

Pitch class distributions can be compared (correlated) with a tonal model to estimate the chords (when considering small time scales) or the key of a piece (when considering a larger time scale). This is the approach followed by Gómez (2004) to estimate the tonality of audio pieces at different temporal scales, as shown on Figure 3.5. To construct the key-profile matrix, Gómez (2004) follows the model for key estimation of MIDI files by Krumhansl (1990). This model considers that tonal hierarchies may be acquired through internalisation of the relative frequencies and durations of tones. The algorithm estimates the key from a set of note duration values, measuring how long each of the 12 pitch classes of an octave (C, C♯, etc.) have been played in a melodic line. In order to estimate the key of the melodic line, the vector of note durations is correlated to a set of key profiles or probe-tone profiles. These profiles represent the tonal hierarchies of the 24 major and minor keys, and each of them contains 12 values, which are the ratings of the degree to which each of the 12 chromatic scale tones fit a particular key. They were obtained by analyzing

human judgments with regard to the relationship between pitch classes and keys (Krumhansl, 1990, pp. 78-81). Gómez (2004) adapts this model to deal with HPCPs (instead of note durations) and polyphonies (instead of melodic lines); details of evaluations can be found in (Gómez, 2006), together with an exhaustive review of computational models of tonality.

3.2.5 Rhythm

Representing rhythm

One way to represent the rhythm of a musical sequence is to specify an exhaustive and accurate list of onset times, maybe together with some other characterizing features such as durations, pitches or intensities (as it is done in MIDI). However, the problem with this representation is the lack of abstraction. There is more to rhythm than the absolute timings of successive music events, one must also consider *tempo*, *meter* and *timing* (Honing, 2001).

Tempo Cooper and Meyer (1960) define a pulse as "[...] one of a series of regularly recurring, precisely equivalent stimuli. [...] Pulses mark off equal units in the temporal continuum." Commonly, "pulse" and "beat" are often used indistinctly and refer both to one element in such a series and to the whole series itself. The tempo is defined as the number of beats in a time unit (usually the minute). There is usually a preferred pulse, which corresponds to the rate at which most people would tap or clap in time with the music. However, the perception of tempo exhibits a degree of variability. It is not always correct to assume that the pulse indicated in a score (Maelzel Metronome) corresponds to the "foot-tapping" rate, nor to the actual "physical tempo" that would be an inherent property of audio streams (Drake et al., 1999). Differences in human perception of tempo depend on age, musical training, music preferences and general listening context (Lapidaki, 1996). They are nevertheless far from random and most often correspond to a focus on a different metrical level and are quantifiable as simple ratios (e.g. 2, 3, $\frac{1}{2}$ or $\frac{1}{3}$).

Meter The metrical structure (or meter) of a music piece is based on the coexistence of several pulses (or "metrical levels"), from low levels (small time divisions) to high levels (longer time divisions). The segmentation of time by a given low-level pulse provides the basic time span to measure music event accentuation whose periodic recurrences define other, higher, metrical levels. The duration-less points in time, the beats, that define this discrete time grid obey a specific set of rules, formalised in the Generative Theory of Tonal Music (Lerdahl and Jackendoff, 1983, GTTM). Beats must be equally spaced. A beat at a high level must also be a beat at each lower level. At any metrical level, a beat which is also a beat at the next higher level is called a downbeat, and other beats are called upbeats. The notions of time signature, measure and bar lines reflect a focus solely on two (or occasionally three) metrical levels. Bar lines define the slower of the two levels (the measure) and the time signature defines the number of faster pulses that make up one measure. For instance, a $\frac{6}{8}$ time signature indicates that the basic temporal unit is an eighth-note and that between two bar lines there is room for six units. Two categories of meter are generally distinguished: duple and triple. This notion is contained in the numerator of the time signature: if the numerator is a multiple of two, then the meter is duple, if it is not a multiple of two but a multiple of three, the meter is triple. The GTTM specifies that there must be a beat of the metrical structure for every note in a music sequence. Accordingly, given a list of note onsets, the quantisation (or "rhythm-parsing") task aims at making it fit into Western music notation. Viable time points (metrical points) are those defined by the different coexisting metrical levels. Quantised durations are then rational numbers (e.g. 1, $\frac{1}{4}$, $\frac{1}{6}$) relative to a chosen time interval: the time signature denominator.

Timing A major weakness of the GTTM is that it does not deal with the deviations from strict metrical timing which occur in almost all styles of music. Thus it is only really suitable for representing the timing structures of music scores, where the expressive timing is not represented. There are conceptually two types of non-metrical timing: long-term tempo deviations (e.g. Rubato) and short-term timing deviations (e.g. "Swing"). One of the greatest difficulties in analyzing performance data is that the two dimensions of expressive timing

are projected onto the single dimension of time. Mathematically, it is possible to represent any tempo change as a series of timing changes and vice-versa, but these descriptions are somewhat counterintuitive (Honing, 2001).

Challenges in automatic rhythm description

Automatic description of musical rhythm is not obvious. First of all because it seems to entail two dichotomic processes: a bottom-up process enabling very rapidly the percept of pulses from scratch, and a top-down process (a persistent mental framework) that lets this induced percept guide the organisation of incoming events (Desain and Honing, 1999). Implementing in a computer program both reactivity to the environment and persistence of internal representations is a challenge. Rhythm description does not solely call for the handling of timing features (onsets and offsets of tones). The definition and understanding of the relationships between rhythm perception and other music features such as intensity or pitches are still open research topics. Rhythm involves two dichotomic aspects that are readily perceived by humans: there are both a strong and complex structuring of phenomena occurring at different time scales and widespread departures from exact metrical timing. Indeed, inexact timings always occur because of expressive performances, sloppy performances and inaccurate collection of timing data (e.g. computational onset detection may have poor time precision and may suffer from false alarms). Furthermore, recent research indicates that even if perceived beats are strongly correlated to onsets of tones, they do not necessarily line up exactly with them, our perception rather favoring smooth tempo curves (Dixon et al., 2006).

Functional framework

The objective of automatic rhythm description is the parsing of acoustic events that occur in time into the more abstract notions of tempo, timing and meter. Algorithms described in the literature differ in their goals. Some of them derive the beats and the tempo of a single metrical level, others try to derive complete rhythmic transcriptions (i.e. musical scores), others aim at determining some timing features from musical performances (such as tempo changes, event

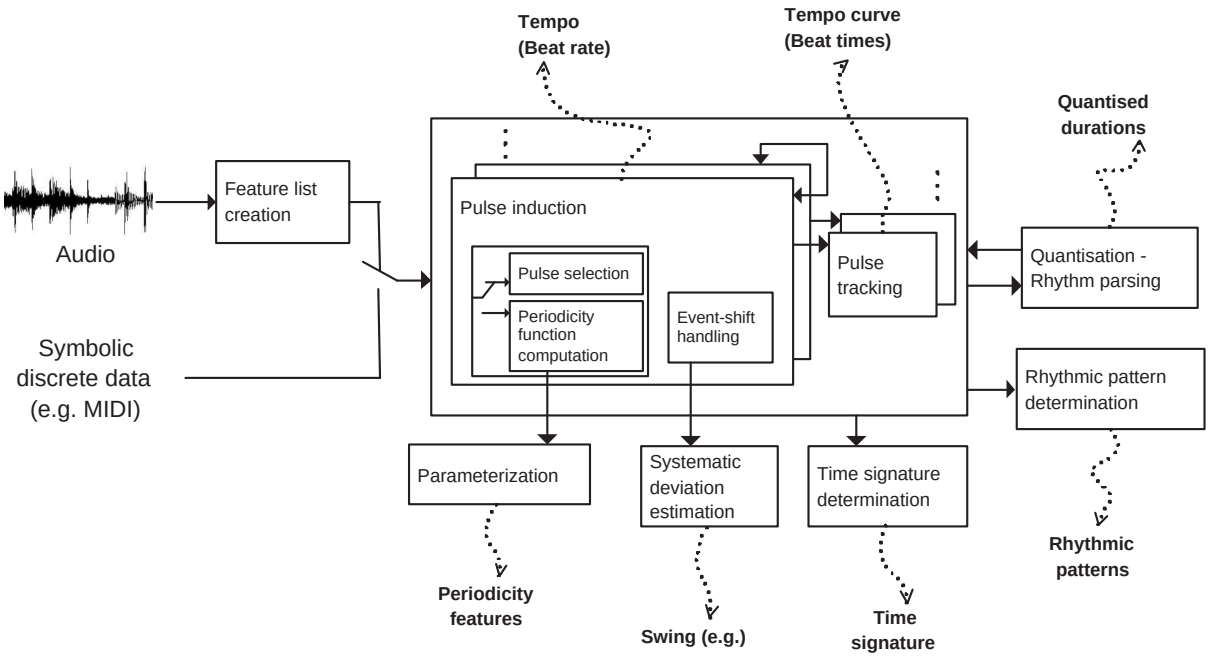

Figure 3.6: Functional units of rhythm description systems.

shifts or swing factors), others focus on the classification of music signals by their overall rhythmic similarities and others look for rhythm patterns. Nevertheless, these computer programs share some functional aspects that we represent as functional blocks of a general diagram in Figure 3.6. We briefly explain each of these functional blocks in the following paragraphs; we refer to the paper by Gouyon and Dixon (2005) for a more complete survey.

Feature list creation Either starting from MIDI, or taking into consideration other symbolic formats such as files containing solely onset times and durations (Brown, 1993), or even raw audio data, the first analysis step is the creation of a feature list, i.e. the parsing, or "filtering," of the data at hand into a sequence that is supposed to convey the predominant information relevant to a rhythmic analysis. These feature lists are defined here broadly, to include frame-based feature vectors as well as lists of symbolic events. The latter include onset times, durations (Brown, 1993), relative amplitude (Dixon, 2001), pitch (Dixon and Cambouropoulos, 2000), chords (Goto, 2001) and percussive

instrument classes (Goto, 2001). Some systems refer to a data granularity of a lower level of abstraction, i.e. at the frame level. Section 3.2.1 describes usual low-level features that can be computed on signal frames. In rhythm analysis, common frame features are energy values and energy values in frequency sub-bands. Some systems also measure energy variations between consecutive frames (Scheirer, 2000; Klapuri et al., 2006). Low-level features other than energy (e.g. spectral flatness, temporal centroid) have also been recently advocated (Gouyon et al., 2006b).

Pulse induction A metrical level (a pulse) is defined by the periodic recurrence of some music event. Therefore, computer programs generally seek periodic behaviors in feature lists in order to select one (or some) pulse period(s) and also sometimes phase(s). This is the process of pulse induction. Concerning pulse induction, computer programs either proceed by *pulse selection*, i.e. evaluating the salience of a restricted number of possible periodicities (Parncutt, 1994), or by computing a periodicity function. In the latter case, a continuous function plots pulse salience versus pulse period (or frequency). Diverse transforms can be used: the Fourier transform, Wavelet transforms, the autocorrelation function, bank of comb filters, etc. In pulse induction, a fundamental assumption is made: the pulse period (and phase) is stable over the data used for its computation. That is, there is no speed variation in that part of the musical performance used for inducing a pulse. In that part of the data, remaining timing deviations (if any) are assumed to be short-time ones (considered as either errors or expressiveness features). They are either "smoothed out," by considering tolerance intervals or smoothing windows, or cautiously handled in order to derive patterns of systematic short-time timing deviations as e.g. the swing. Another step is necessary to output a discrete pulse period (and optionally its phase) rather than a continuous periodicity function. This is usually achieved by a peak-picking algorithm.

Pulse tracking Pulse tracking and pulse induction often occur as complementary processes. Pulse induction models consider short term timing deviations as noise, assuming a relatively stable tempo, whereas a pulse tracker

handles the short term timing deviations and attempts to determine changes in the pulse period and phase, without assuming that the tempo remains constant. Another difference is that induction models work bottom-up, whereas tracking models tend to follow top-down approaches, driven by the pulse period and phase computed in a previous induction step. Pulse tracking is often a process of reconciliation between predictions (driven by previous period and phase computations) and the observed data. Diverse formalisms and techniques have been used in the design of pulse trackers: rule-based (Desain and Honing, 1999), problem-solving (Allen and Dannenberg, 1990), agents (Dixon, 2001), adaptive oscillators (Large and Kolen, 1994), dynamical systems (Cemgil et al., 2001), Bayesian statistics (Raphael, 2002) and particle filtering (Hainsworth and Macleod, 2004). A complete review can be found in the already mentioned paper by Gouyon and Dixon (2005). Some systems rather address pulse tracking by "repeated induction" (Scheirer, 2000; Laroche, 2003; Klapuri et al., 2006). A pulse is induced on a short analysis window (e.g. around 5 seconds of data), then the window is shifted in time and another induction takes place. Determining the tempo evolution then amounts to connecting the observations at each step. In addition to computational overload, one problem that arises with this approach to pulse tracking is the lack of continuity between successive observations and the difficulty of modelling sharp tempo changes.

Quantisation and time signature determination Few algorithms for time signature determination exist. The simplest approach is based on parsing the peaks of the periodicity function to find two significant peaks, which correspond respectively to a fast pulse, the time signature denominator, and a slower pulse, the numerator (Brown, 1993). The ratio between the pulse periods defines the time signature. Another approach is to consider all pairs of peaks as possible beat/measure combinations, and compute the fit of all periodicity peaks to each hypothesis (Dixon et al., 2003). Another strategy is to break the problem into several stages: determining the time signature denominator (e.g. by tempo induction and tracking), segmenting the music data with respect to this pulse and compute features at this temporal scope and finally detecting periodicities in the created feature lists (Gouyon and Herrera, 2003). Quantisation (or "rhythm parsing") can be seen as a by-product of

the induction of several metrical levels, which together define a metrical grid. The rhythm of a given onset sequence can be parsed by assigning each onset (independently of its neighbors) to the closest element in this hierarchy. The weaknesses of such an approach are that it fails to account for musical context (e.g. a triplet note is usually followed by 2 more) and deviations from the metrical structure. Improvements to this first approach are considered by Desain and Honing (1989). Arguing that deviations from the metrical structure would be easier to determine if the quantised durations were known (Allen and Dannenberg, 1990), many researchers now consider rhythm parsing simultaneously with tempo tracking (Raphael, 2002; Cemgil and Kappen, 2003), rather than subsequent to it (hence the bi-directional arrow between these two modules in Figure 3.6).

Systematic deviation characterisation In the pulse induction process, short-term timing deviations can be "smoothed out" or cautiously handled so as to derive patterns of short-term timing deviations, such as swing: a "long-short" timing pattern of consecutive eight-notes. For instance, Laroche (2001) proposes to estimate the swing jointly with tempo and beats at the half-note level, assuming constant tempo: all pulse periods, phases and eight-note "long-short" patterns are enumerated and a search procedure determines which one best matches the onsets.

Rhythmic pattern determination Systematic short-term timing deviations are important music features. In addition, repetitive rhythmic patterns covering a longer temporal scope can also be characteristic of some music styles. For instance, many electronic musical devices feature templates of prototypical patterns such as Waltz, Cha Cha and the like. The length of such patterns is typically one bar, or a couple or bars. Few algorithms have been proposed for the automatic extraction of rhythmic patterns; they usually require the knowledge (or previous extraction) of part of the metrical structure, typically the beats and measure (Dixon et al., 2004).

Periodicity features Other rhythmic features, with a musical meaning less explicit than for example the tempo or the swing, have also been proposed, in particular in the context of designing rhythm similarity distances. Most of the time, these features are derived from a parametrisation of a periodicity function, e.g. the salience of several prominent peaks (Gouyon et al., 2004), their positions (Tzanetakis and Cook, 2002; Dixon et al., 2003), selected statistics (high-order moments, flatness, etc.) of the periodicity function considered as a probability density function (Gouyon et al., 2004) or simply the whole periodicity function itself (Foote et al., 2002).

Future research directions

Current research in rhythm description addresses all of these aspects, with varying degrees of success. For instance, determining the tempo of music with minor speed variations is feasible for almost all music styles if we do not expect that the system finds a specific metrical level (Gouyon et al., 2006a). Recent pulse tracking systems also reach high levels of accuracy. On the other hand, accurate quantisation, score transcription, determination of time signature and characterisation of intentional timing deviations are still open question. Particularly, it remains to be investigated how general recently proposed models are with respect to different music styles. New research directions include the determination of highly abstract rhythmic features required for music content processing and music information retrieval applications, the definition of the best rhythmic features and the most appropriate periodicity detection method (Gouyon, 2005).

3.2.6 Genre

Most music can be described in terms of dimensions such as melody, harmony, rhythm, etc. These high-level features characterise music and at least partially determine its genre, but, as mentioned in previous sections, they are difficult to compute automatically from raw audio signals. As a result, most audio-related music information retrieval research has focused on low-level features

and induction algorithms to perform genre classification tasks. This approach has met with some success, but it is limited by the fact that the low level of representation may conceal many of the truly relevant aspects of a piece of music. See Chapter 4 and the works by Pampalk (2006), Ahrendt (2006), and Aucouturier (2006) for reviews of the current state-of-the-art in genre classification and more information on promising directions.

3.3 Audio content exploitation

We consider in this section a number of applications of content-based descriptions of audio signals. Although audio retrieval (see Section 3.3.1) is the one that has been addressed most often, others deserve a mention, e.g. content-based transformations (see Section 3.3.2).

3.3.1 Content-based search and retrieval

Searching a repository of music pieces can be greatly facilitated by automatic description of audio and music content (Cano, 2007), e.g. fingerprints, melodic features, tempo, etc. A content-based music retrieval system is a search engine at the interface of a repository, or organised database, of music pieces. Typically,

1. it receives a query, defined by means of musical strategies (e.g. humming, tapping, providing an audio excerpt or some measures of a score) or textual strategies (e.g. using "words" and/or "numbers" that describe some music feature like tempo, mood, etc.) referring to audio or music descriptors;

2. it has access to the set of music features extracted from the music files in the repository;

3. it returns a list of ranked files or excerpts that

   (a) are all relevant to the query (i.e. with high precision) or

(b) constitute the set of all relevant files in the database (i.e. high recall);

4. (optionally) it processes some user-feedback information in order to improve its performance in the future.

Identification

With the help of fingerprinting systems it is possible to identify an unlabelled piece of audio and therefore provide a link to corresponding metadata (e.g. artist and song name). Depending on the application, different importance may be given to the following requirements:

Accuracy: The number of correct identifications, missed identifications, and wrong identifications (false positives).

Reliability: This is of major importance for copyright enforcement organisations.

Robustness: Ability to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel. Other sources of degradation are pitching, equalisation, background noise, D/A-A/D conversion, audio coders (such as GSM and MP3), etc.

Granularity: Ability to identify whole titles from excerpts a few seconds long. It needs to deal with shifting, that is lack of synchronisation between the extracted fingerprint and those stored in the database and it adds complexity to the search (it needs to compare audio in all possible alignments).

Security: Vulnerability of the solution to cracking or tampering. In contrast with the robustness requirement, the manipulations to deal with are designed to fool the fingerprint identification algorithm.

Versatility: Ability to identify audio regardless of the audio format. Ability to use the same database for different applications.

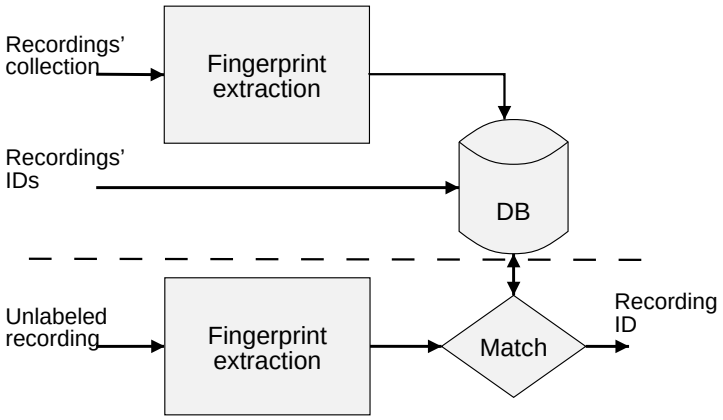

Figure 3.7: Content-based audio identification framework

Scalability: Performance with very large databases of titles or a large number of concurrent identifications. This affects the accuracy and the complexity of the system.

Complexity: It refers to the computational costs of the fingerprint extraction, the size of the fingerprint, the complexity of the search, the complexity of the fingerprint comparison, the cost of adding new items to the database, etc.

Fragility: Some applications, such as content-integrity verification systems, may require the detection of changes in the content. This is contrary to the robustness requirement, as the fingerprint should be robust to content-preserving transformations but not to other distortions.

The requirements of a complete fingerprinting system should be considered together with the fingerprint requirements listed in Section 3.2.3. Bear in mind that improving a certain requirement often implies loosing performance in some other. The overall identification process mimics the way humans perform the task. As seen in Figure 3.7, a memory of the recordings to be recognised is created off-line (top); in the identification mode (bottom), unlabelled audio is presented to the system to look for a match.

Audio Content Monitoring and Tracking One of the commercial usages of audio identification is that of remotely controlling the times a piece of music has been broadcasted, in order to ensure the broadcaster is doing the proper clearance for the rights involved (Cano et al., 2002).

Monitoring at the distributor end Content distributors may need to know whether they have the rights to broadcast certain content to consumers. Fingerprinting helps identify unlabelled audio in TV and radio channels' repositories. It can also identify unidentified audio content recovered from CD plants and distributors in anti-piracy investigations (e.g. screening of master recordings at CD manufacturing plants).

Monitoring at the transmission channel In many countries, radio stations must pay royalties for the music they air. Rights holders are eager to monitor radio transmissions in order to verify whether royalties are being properly paid. Even in countries where radio stations can freely air music, rights holders are interested in monitoring radio transmissions for statistical purposes. Advertisers are also willing to monitor radio and TV transmissions to verify whether commercials are being broadcast as agreed. The same is true for web broadcasts. Other uses include chart compilations for statistical analysis of program material or enforcement of cultural laws (e.g. in France, a certain percentage of the aired recordings must be in French). Fingerprinting-based monitoring systems can be used for this purpose. The system "listens" to the radio and continuously updates a play list of songs or commercials broadcast by each station. Of course, a database containing fingerprints of all songs and commercials to be identified must be available to the system, and this database must be updated as new songs come out. Examples of commercial providers of such services are: `http://www.musicreporter.net`, `http://www.audiblemagic.com`, `http://www.yacast.fr`, `http://www.musicip.com/` and `http://www.bmat.com/`. Additionally, audio content can be found in web pages and web-based peer-to-peer networks. Audio fingerprinting combined with a web crawler can identify their content and report it to the corresponding rights owners (e.g. `http://www.baytsp.com`).

Monitoring at the consumer end In usage-policy monitoring applications, the goal is to avoid misuse of audio signals by the consumer. We can conceive a system where a piece of music is identified by means of a fingerprint and a database is contacted to retrieve information about the rights. This information dictates the behavior of compliant devices (e.g. CD and DVD players and recorders, MP3 players or even computers) in accordance with the usage policy. Compliant devices are required to be connected to a network in order to access the database.

Added-value services Some systems store metadata related to audio files in databases accessible through the Internet. Such metadata can be relevant to a user for a given application and covers diverse types of information related to an audio file (e.g. how it was composed and how it was recorded, the composer, year of composition, the album cover image, album price, artist biography, information on the next concerts, etc.). Fingerprinting can then be used to identify a recording and retrieve the corresponding metadata. For example, MusicBrainz (`http://www.musicbrainz.org`), Id3man (`http://www.id3man.com`) or Moodlogic (`http://www.moodlogic.com`) automatically label collections of audio files. The user can download a compatible player that extracts fingerprints and submits them to a central server from which metadata associated to the recordings is downloaded. Gracenote (`http://www.gracenote.com`) recently enhanced their technology based on CDs' tables of contents with audio fingerprinting. Another application consists in finding or buying a song while it is being broadcast, by means of mobile-phone transmitting its GPS-quality received sound (e.g. `http://www.shazam.com`, `http://www.bmat.com`).

Summarisation

Summarisation, or thumbnailing, is essential for providing fast-browsing functionalities to content processing systems. An audiovisual summary that can be played, skipped upon, replayed or zoomed can save the user time and help him/her to get a glimpse of "what the music is about," especially when using personal media devices. Music summarisation consists in determining the

key elements of a music sound file and rendering them in the most efficient way. There are two tasks here: first, extracting structure (Ong, 2007), and then, creating aural and visual representations of this structure (Peeters et al., 2002). Extracting a good summary from a sound file needs a comprehensive description of its content, plus some perceptual and cognitive constraints to be derived from users. An additional difficulty here is that different types of summaries can coexist, and that different users will probably require different summaries. Because of this amount of difficulty, the area of music summarisation is still under-developed. Reviews of recent promising approaches are presented by Ong (2007).

Play-list generation

This area concerns the design of lists of music pieces that satisfy some ordering criteria with respect to content descriptors previously computed, indicated (explicitly or implicitly) by the listener (Pampalk and Gasser, 2006). Play-list generation is usually constrained by time-evolving conditions (e.g. "start with slow-tempo pieces, then progressively increase tempo") (Pachet et al., 2000). Besides the play-list construction problem, we can also mention related problems such as achieving seamless transitions (in user-defined terms such as tempo, tonality, loudness) between the played pieces.

Music browsing and recommendation

Music browsing and recommendation are very much demanded, especially among youngsters. Recommendation consists in suggesting, providing guidance, or advising a potential consumer about interesting music files in on-line music stores, for instance. Nowadays, this is mainly possible by querying artist or song names (or other types of editorial data such as genre), or by browsing recommendations generated by collaborative filtering, i.e. using recommender systems that exploit information of the type "users that bought this album also bought this album." An obvious drawback of the first approach is that consumers need to know the name of the song or the artist beforehand. The second approach is only suitable when a considerable number

of consumers has heard and rated the music. This situation makes it difficult for users to access and discover the vast amount of music composed and performed by unknown artists which is available in an increasing number of sites (e.g. `http://www.magnatune.com`) and which nobody has yet rated nor described. Content-based methods represent an alternative to these approaches. See the paper by Cano et al. (2005c) for the description of a large-scale music browsing and recommendation system based on automatic description of music content.[8] Other approaches to music recommendation are based on users' profiles: users' musical tastes and listening habits as well as complementary contextual (e.g. geographical) information (Celma, 2006a).[9] It is reasonable to assume that these different approaches will merge in the near future and result in improved music browsing and recommendation systems (Celma, 2006b).

Content visualisation

The last decade has witnessed great progress in the field of data visualisation. Massive amounts of data can be represented in multidimensional graphs in order to facilitate comparisons, grasp the patterns and relationships between data, and improve our understanding of them. Four purposes of information visualisation can be distinguished (Hearst, 1999):

Exploration, where visual interfaces can also be used as navigation and browsing interfaces.

Computation, where images are used as tools for supporting the analysis and reasoning about information. Data insight is usually facilitated by good data visualisations.

Communication, where images are used to summarise what otherwise would need many words and complex concepts to be understood. Music visualisation tools can be used to present concise information about relationships extracted from many interacting variables.

[8] See also `http://musicsurfer.iua.upf.edu`
[9] see `http://foafing-the-music.iua.upf.edu`

Decoration, where content data are used to create attractive pictures whose primary objective is not the presentation of information but aesthetic amusement.

It is likely that in the near future we will witness an increasing exploitation of data visualisation techniques in order to enhance song retrieval, collection navigation and music discovery (Pampalk and Goto, 2006).

3.3.2 Content-based audio transformations

Transformations of audio signals have a long tradition (Zölzer, 2002). A recent trend in this area of research is the editing and transformation of music audio signals triggered by explicit musically-meaningful representational elements, in contrast to low-level signal descriptors. These techniques are referred to as content-based audio transformations, or "adaptive digital audio effects" (Verfaille et al., 2006), and are based on the type of description of audio signals detailed above in Section 3.2. In this section, we give examples of such techniques, following increasing levels of abstraction in the corresponding content description.

Loudness modifications

The most commonly known effects related to loudness are the ones that modify the sound intensity level: volume change, tremolo, compressor, expander, noise gate and limiter (Verfaille et al., 2006). However, when combined with other low-level features, loudness is correlated to higher-level descriptions of sounds, such as the timbre or the musical intentions of a performer. It can therefore be used as a means to control musically-meaningful aspects of sounds. The mechanisms that relate the actions of a player to the sound level produced by a given instrument are usually so complex that this feature can seldom be decorrelated from others, such as timbre. Thus, differences between playing a soft and a loud note on an instrument do not reside only in loudness levels. Spectral modifications must also be accounted for. In the case of the singing voice, for instance, many studies have been carried out and are

summarised by Sundberg (1987). Using Sundberg's nomenclature, it is possible, under certain conditions, to infer the source spectrum modifications from uttering the same vowel at different loudness of phonation. Building upon this assumption, Fabig and Janer (2004) propose a method for modifying the loudness of the singing voice by detecting the excitation slope automatically.

Time-scaling

In a musical context, time-scaling can be understood as changing the pace of a music signal, i.e. its tempo. If a musical performance is time-scaled to a different tempo, we should expect to listen to the same notes starting at a scaled time pattern, but with durations modified linearly according to the tempo change. The pitch of the notes should however remain unchanged, as well as the perceived expression. Thus, for example, vibratos should not change their depth, tremolo or rate characteristics. And of course, the audio quality should be preserved in such a way that if we had never listened to that music piece, we would not be able to know if we were listening to the original recording or to a transformed one. Time-scale modifications can be implemented in different ways. Generally, algorithms are grouped in three different categories: time domain techniques, phase-vocoder and variants, and signal models. In the remainder of this section we explain the basics of these approaches in turn.

Time domain techniques Time domain techniques are the simplest methods for performing time-scale modification. The simplest (and historically first) technique is the variable speed replay of analog audio tape recorders (McNally, 1984). A drawback of this technique is that during faster playback, the pitch of the sound is raised while the duration is shortened. On the other hand, during slower playback, the pitch of the sound is lowered while the duration is lengthened. Many papers show good results without scaling frequency by segmenting the input signal into several windowed sections and then placing these sections in new time locations and overlapping them to get the time-scaled version of the input signal. This set of algorithms is referred to as

Overlap-Add (OLA). To avoid phase discontinuities between segments, the synchronised OLA algorithm (SOLA) uses a cross-correlation approach to determine where to place the segment boundaries (Wayman et al., 1989). In TD-PSOLA (Moulines et al., 1989), the overlapping operation is performed pitch-synchronously to achieve high quality time-scale modification. This works well with signals having a prominent basic frequency and can be used with all kinds of signals consisting of a single signal source. When it comes to a mixture of signals, this method will produce satisfactory results only if the size of the overlapping segments is increased to include a multiple of cycles, thus averaging the phase error over a longer segment and making it less audible. WSOLA (Verhelst and Roelands, 1993) uses the concept of waveform similarity to ensure signal continuity at segment joints, providing high quality output with high algorithmic and computational efficiency and robustness. All the aforementioned techniques consider equally the transient and steady state parts of the input signal, and thus time-scale them both in the same way. To get better results, it is preferable to detect the transient regions and not time-scale them, just translate them into a new time position, while time-scaling the non-transient segments. The earliest mention of this technique can be found in the *Lexicon 2400* time compressor/expander from 1986. This system detects transients, and time-scales only the remaining audio using a TD-PSOLA-like algorithm. Lee et al. (1997) show that using time-scale modification on non-transient parts of speech alone improves the intelligibility and quality of the resulting time-scaled speech.

Phase vocoder and variants The phase-vocoder is a relatively old technique that dates from the 70's (Portnoff, 1976). It is a frequency domain algorithm computationally quite more expensive than time domain algorithms. However it can achieve high-quality results even with high time-scale factors. Basically, the input signal is split into many frequency channels, uniformly spaced, usually using the FFT. Each frequency band (bin) is decomposed into magnitude and phase parameters, which are modified and re-synthesised by the IFFT or a bank of oscillators. With no transformations, the system allows a perfect reconstruction of the original signal. In the case of time-scale modification, the synthesis hop size is changed according to the desired time-scale factor.

Magnitudes are linearly interpolated and phases are modified in such a way that phase consistency are maintained across the new frame boundaries. The phase-vocoder introduces signal smearing for impulsive signals due to the loss of phase alignment of the partials. A typical drawback of the phase vocoder is the loss of vertical phase coherence that produces reverberation or loss of presence in the output. This effect is also referred to as "phasiness," which can be circumvented by phase-locking techniques (Laroche and Dolson, 1999) among bins around spectral peaks. Note that adding peak tracking to the spectral peaks, the phase-vocoder resembles the sinusoidal modelling algorithms, which is introduced in the next paragraph. Another traditional drawback of the phase vocoder is the bin resolution dilemma: the phase estimates are incorrect if more than one sinusoidal peak resides within a single spectral bin. Increasing the window may solve the phase estimation problem, but it implies a poor time resolution and smooths the fast frequency changes. And the situation gets worse in the case of polyphonic music sources because then the probability is higher that sinusoidal peaks from different sources will reside in the same spectrum bin. Different temporal resolutions for different frequencies can be obtained by convolution of the spectrum with a variable kernel function (Hoek, 1999). Thus, long windows are used to calculate low frequencies, while short windows are used to calculate high frequencies. Other approaches approximate a constant-Q phase-vocoder based on wavelet transforms or non-uniform sampling.

Techniques based on signal models Signal models have the ability to split the input signal into different components which can be parameterised and processed independently giving a lot of flexibility for transformations. Typically these components are sinusoids, transients and noise. In sinusoidal modelling (McAulay and Quatieri, 1986), the input signal is represented as a sum of sinusoids with time-varying amplitude, phase and frequency. Parameter estimation can be improved by using interpolation methods, signal derivatives and special windows. Time-scaling using sinusoidal modelling achieves good results with harmonic signals, especially when keeping the vertical phase coherence. However it fails to successfully represent and transform noise and transient signals. Attacks are smoothed and noise sounds artificial.

The idea of subtracting the estimated sinusoids from the original sound to obtain a residual signal was proposed by Smith and Serra (1987); this residual can then be modelled as a stochastic signal. This method allows the splitting of e.g. a flute sound into the air flow and the harmonics components, and to transform both parts independently. This technique successfully improves the quality of time-scale transformations but fails to handle transients, which are explicitly handled in (Verma et al., 1997). Then, all three components (sinusoidal, noise and transient) can be modified independently and re-synthesised. When time-scaling an input signal, transients can successfully be translated to a new onset location, preserving their perceptual characteristics.

Timbre modifications

Timbre is defined as all those characteristics that distinguish two sounds having the same pitch, duration and loudness. As a matter of fact, timbre perception depends on many characteristics of the signal such as its instantaneous spectral shape and its evolution, the relation of its harmonics, and some other features related to the attack, release and temporal structure. Timbre instrument modification can be achieved by many different techniques. One of them is to modify the input spectral shape by *timbre mapping*. Timbre mapping is a general transformation performed by warping the spectral shape of a sound by means of a mapping function $g(f)$ that maps frequencies of the transformed spectrum (f_y) to frequencies of the initial spectrum (f_x) via a simple equation $f_y = g(f_x)$. Linear *scaling* (compressing or expanding) is a particular case of timbre mapping in which the mapping function pertains to the family $f_y = k * f_x$, where k is the scale factor, usually between 0.5 and 2. The timbre scaling effect resembles modifications of the size and shape of the instrument. The *shifting* transformation is another particular case of the timbre mapping as well, in which $g(f)$ can be expressed as $f_y = f_x + c$, where c is an offset factor.

Morphing Another way of accomplishing timbre transformations is to modify the input spectral shape by means of a secondary spectral shape. This is usually referred to as *morphing* or *cross-synthesis*. In fact, morphing is a

technique with which, out of two or more elements, we can generate new ones with hybrid properties. In the context of video processing, morphing has been widely developed and enjoys great popularity in commercials, video clips and films where faces of different people change one into another or chairs mutate into for example elephants. Analogously, in the context of audio processing, the goal of most of the developed morphing methods has been the smooth transformation from one sound to another. Along this transformation, the properties of both sounds combine and merge into a resulting hybrid sound. With different names, and using different signal processing techniques, the idea of audio morphing is well known in the computer music community (Serra, 1994; Slaney et al., 1996). In most algorithms, morphing is based on the interpolation of sound parameterisations resulting from analysis/synthesis techniques, such as the short-time Fourier transform, linear predictive coding or sinusoidal models.

Voice timbre Whenever the morphing is performed by means of modifying a reference voice signal in matching its individuality parameters to another, we can refer to it as voice conversion (Loscos, 2007). Some applications for the singing voice exist in the context of karaoke entertainment (Cano et al., 2000) and in the related topics of gender change (Cano et al., 2000) and unison choir generation (Bonada et al., 2006). We refer to the paper by Bonada and Serra (2007) regarding the general topic of singing voice synthesis. Still for the particular case of voice, other finer-grained transformations exist to modify the timbre character without resorting to a morphing between two spectral shapes: e.g. *rough*, *growl*, *breath* and *whisper* transformations. *Roughness* in voice can come from different pathologies such as biphonia, or diplophonia, and can combine with many other voice tags such as "hoarse" or "creaky." However, here we will refer to a rough voice as the one due to cycle to cycle variations of the fundamental frequency (jitter), and the period amplitude (shimmer). The most common techniques used to synthesise rough voices work with a source/filter model and reproduce the jitter and shimmer aperiodicities in the time domain (Childers, 1990). These aperiodicities can be applied to the voiced pulse-train excitation by taking real patterns that have been extracted from rough voice recordings or by using statistical models (Schoentgen, 2001).

Spectral domain techniques have also proved to be valid to emulate roughness (Loscos and Bonada, 2004). *Growl* phonation is often used when singing jazz, blues, pop and other music styles as an expressive accent. Perceptually, growl voices are close to other dysphonic voices such as "hoarse" or "creaky," however, unlike these others, growl is always a vocal effect and not a permanent vocal disorder. According to Sakakibara et al. (2004), growl comes from simultaneous vibrations of the vocal folds and supra-glottal structures of the larynx. The vocal folds vibrate half periodically to the aryepiglottic fold vibration generating sub-harmonics. Growl effect can be achieved by adding these sub-harmonics in frequency domain to the original input voice spectrum (Loscos and Bonada, 2004). These sub-harmonics follow certain magnitude and phase patterns that can be modelled from spectral analyses of real growl voice recordings. *Breath* can be achieved by different techniques. One is to increase the amount of the noisy residual component in those sound models in which there is a sinusoidal-noise decomposition. For sound models based on the phase-locked vocoder (see Section 3.3.2) a more breathy timbre can be achieved by filtering and distorting the harmonic peaks. The *whisper* effect can be obtained by equalizing a previously recorded and analyzed template of a whisper utterance. The time behavior of the template is preserved by adding to the equalisation the difference between the spectral shape of the frame of the template currently being used and an average spectral shape of the template. An "anti-proximity" filter may be applied to achieve a more natural and smoother effect (Fabig and Janer, 2004).

Rhythm transformations

In addition to tempo changes (see Section 3.3.2), some existing music editing softwares provide several rhythm transformation functionalities. For instance, any sequencer provides the means to adjust MIDI note timings to a metrical grid ("quantisation") or a predefined rhythmic template. By doing an appropriate mapping between MIDI notes and audio samples, it is therefore possible to apply similar timing changes to audio mixes. But when dealing with general polyphonic music excerpts, without corresponding MIDI scores, these techniques cannot be applied. Few commercial applications implement

techniques to transform the rhythm of general polyphonic music excerpts. A review can be found in the paper by Gouyon et al. (2003a). A technique for swing transformation has also been proposed in the same paper by Gouyon et al. (2003a), which consists of a description module and a transformation module. The description module does onset detection and rhythmic analysis (see Section 3.2.5). Swing is relative to the length of consecutive eighth-notes, it is therefore necessary to determine the beat indexes of eighth-notes. It is also necessary to describe the excerpt at the next higher (slower) metrical level, the quarter-note, and determine the eighth-note "phase," that is, determine in a group of two eighth-notes which is the first one.[10] The existing ratio between consecutive eighth-notes is also estimated. This ratio can be changed by shortening or lengthening the first eighth-notes of each quarter-note, and lengthening or shortening the second eighth-notes accordingly. This is done by means of time-scaling techniques. In the papers by Gouyon et al. (2003a) and Janer et al. (2006), time-scaling is done in real-time and the user can continuously adjust the swing ratio while playing back the audio file. Having found evidence for the fact that deviations occurring within the scope of the smallest metrical pulse are very important for musical expressiveness, Bilmes (1993) proposes additional rhythmic transformations based on a high-level description of the rhythmic content of audio signals. Interesting recent applications of rhythm transformations can be found in the works by Wright and Berdahl (2006), Ramirez and Hazan (2006), Janer et al. (2006), Grachten et al. (2006), and Ravelli et al. (2007).

Melodic transformations

Melodic transformations such as *pitch discretisation to temperate scale* and *intonation* apply direct modifications to the fundamental frequency envelope. Arguably, these transformations may be considered low level transformations; however, they do change the way a high-level descriptor, namely the melody, is perceived by the listener. Intonation transformations are achieved by stretching or compressing the difference between the analysis pitch envelope and a

[10] Indeed, it is not at all the same to perform a "long-short" pattern as a "short-long" pattern.

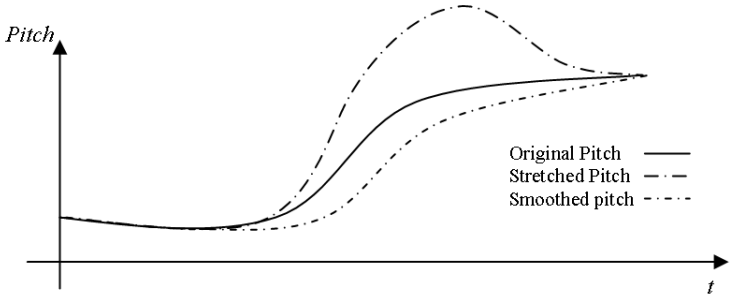

Figure 3.8: Intonation transformation.

low pass filtered version of it. The goal of the transformation is to increase or decrease the sharpness of the note attack, as illustrated in Figure 3.8. Pitch discretisation to temperate scale can be accomplished by forcing the pitch to take the nearest frequency value of the equal temperate scale. It is indeed a very particular case of pitch transposition where the pitch is quantified to one of the 12 semitones of an octave (Amatriain et al., 2003).[11] Other melodic transformations can be found in the software *Melodyne*[12] and in the paper by Gómez et al. (2003d) as *transposition* (global change of pitch), *horizontal symmetry*, in which the user can choose a pitch value (arbitrary or some global descriptor related to pitch distribution as minimum, maximum or mean pitch value of the melody) and perform a symmetric transformation of the note pitches with respect to this value on a horizontal axis, *contour direction changes* in which the user can change the interval direction without changing the interval depth (e.g. converting an ascending octave to a descending one), etc. Although these transformations are conceptually simple, they correspond to usual music composition procedures and can create dramatic changes that may enhance the original material (if used in the right creative context). Finally, melodies of monophonic instruments can also be transformed by applying changes on other high-level descriptors in addition to the pitch, such as tempo curves (Grachten et al., 2006) and note timing and loudness (Ramirez et al., 2004). See Chapter 5 for more information on analysis and generation of expressive musical performances.

[11] See also Antares' *Autotune*, `http://www.antarestech.com/`.
[12] `http://www.celemony.com/melodyne`

Harmony transformations

Harmonizing a sound can be defined as mixing a sound with several pitch-shifted versions of it (Amatriain et al., 2002; Verfaille et al., 2006). This requires two parameters: the number of voices and the pitch for each of these. Pitches of the voices to generate are typically specified by the key and chord of harmonisation. Where the key and chord are estimated from the analysis of the input pitch and the melodic context (Pachet and Roy, 1998), some refer to "intelligent harmonizing."[13] An application of harmonizing in real-time monophonic solo voices is detailed in (Bonada et al., 2006).

3.4 Perspectives

All areas of high level description of music audio signals (as for instance those addressed in this chapter – Tonality, Rhythm, etc. –) will, without doubt, witness rapid improvements in the near future. We believe however that the critical aspect to focus on for these improvements to happen is the systematic use of large-scale evaluations.

Evaluations Developing technologies related to content processing of music audio signals requires data (Cano et al., 2004). For instance, implementing algorithms for automatic instrument classification requires annotated samples of different instruments. Implementing a voice synthesis and transformation software calls for repositories of voice excerpts sung by professional singers. Testing a robust beat-tracking algorithm requires songs of different styles, instrumentation and tempi. Building models of music content with a Machine Learning rationale calls for large amounts of data. Besides, running an algorithm on big amounts of (diverse) data is a requirement to ensure the algorithm's quality and reliability. In other scientific disciplines long-term improvements have shown to be bounded to systematic evaluation of models. For instance, text retrieval techniques significantly improved over the year thanks

[13] see TC-Helicon's *Voice Pro*, `http://www.tc-helicon.com/VoicePro`.

to the TREC initiative (see http://trec.nist.gov). TREC evaluations proceed by giving research teams access to a standardised, large-scale test collection of text, a standardised set of test queries, and requesting a standardised way of generating and presenting the results. Different TREC tracks have been created over the years (text with moving images, web retrieval, speech retrieval, etc.) and each track has developed its own special test collections, queries and evaluation requirements. The standardisation of databases and evaluation metrics also greatly facilitated progress in the fields of Speech Recognition (Przybocki and Martin, 1989; Pearce and Hirsch, 2000), Machine Learning (Guyon et al., 2004) or Video Retrieval (see http://www-nlpir.nist.gov/projects/trecvid/). In 1992, the visionary Marvin Minsky declared: "the most critical thing, in both music research and general AI research, is to learn how to build a common music database" (Minsky and Laske, 1992). More than 10 years later, this is still an open issue. In the last few years, the music content processing community has recognised the necessity of conducting rigorous and comprehensive evaluations (Downie, 2002, 2003b). However, we are still far from having set a clear path to be followed for evaluating research progresses. Inspired by Downie (2003b), here follows a list of urgent methodological problems to be addressed by the research community:

1. There are no standard collections of music against which to test content description or exploitation techniques;
2. There are no standardised sets of performance tasks;
3. There are no standardised evaluation metrics.

As a first step, an audio description contest took place during the fifth edition of the ISMIR, in Barcelona, in October 2004. The goal of this contest was to compare state-of-the-art audio algorithms and systems relevant for some tasks of music content description, namely genre recognition, artist identification, tempo extraction, rhythm classification and melody extraction (Cano et al., 2006b). It was the first large-scale evaluation of audio description algorithms, and the first initiative to make data and legacy metadata publicly available (see http://ismir2004.ismir.net and Cano et al., 2006b, for more details). However, this competition addressed a small part of the bulk of research going on

in music content processing. Following editions of the ISMIR have continued this effort: public evaluations now take place on an annual basis, in the Music Information Retrieval Evaluation eXchange (MIREX), organised during ISMIR conferences mainly by the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL)[14] together with voluntary fellow researchers. MIREXes have widened the scope of the competitions and cover a broad range of tasks, including symbolic data description and retrieval. Future editions of MIREX are likely to make a further step, from evaluation of content description algorithms to evaluations of complete MIR systems.

Acknowledgments

This work has been partially funded by the European IST-507142 project SIMAC (Semantic Interaction with Music Audio Contents),[15] and the HARMOS E-Content project. The authors wish to thank their colleagues in the Music Technology Group for their help. Thanks also to Simon Dixon for participating in previous versions of Section 3.2.5.

[14] http://www.music-ir.org/evaluation/
[15] http://www.semanticaudio.org

Bibliography

P. Ahrendt. *Music Genre Classification Systems - A Computational Approach*. Unpublished PhD thesis, Technical University of Denmark, 2006.

P. Aigrain. New applications of content processing of music. *Journal of New Music Research*, 28(4):271–280, 1999.

P. Allen and R. Dannenberg. Tracking musical beats in real time. In Allen P. and Dannenberg R., editors, *Proceedings International Computer Music Conference*, pages 140–143, 1990.

X. Amatriain, J. Bonada, À. Loscos, and X. Serra. Spectral processing. In U. Zölzer, editor, *DAFX Digital Audio Effects*, pages 373–439. Wiley & Sons, 2002.

X. Amatriain, J. Bonada, À. Loscos, J. Arcos, and V. Verfaille. Content-based transformations. *Journal of New Music Research*, 32(1):95–114, 2003.

J-J. Aucouturier. *Ten Experiments on the Modelling of Polyphonic Timbre*. Unpublished PhD thesis, University of Paris 6, Paris, 2006.

M. Basseville and I. V. Nikiforov. *Detection of abrupt changes: Theory and application*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1993.

E. Batlle and P. Cano. Automatic segmentation for music classification using competitive hidden markov models. In *Proceedings International Symposium on Music Information Retrieval*, 2000.

E. Batlle, J. Masip, and E. Guaus. Automatic song identification in noisy broadcast audio. In *Proceedings of IASTED International Conference on Signal and Image Processing*, 2002.

J. Bello. *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach*. Unpublished PhD thesis, Department of Electronic Engineering, Queen Mary University of London, 2003.

J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

J. Bilmes. Techniques to foster drum machine expressivity. In *Proceedings International Computer Music Conference*, pages 276–283, 1993.

S. Blackburn. *Content based retrieval and navigation of music using melodic pitch contour*. Unpublished PhD thesis, University of Southampton, 2000.

T. Blum, D. Keislar, J. Wheaton, and E. Wold. Method and article of manufacture for content-based analysis, storage, retrieval and segmentation of audio information, U.S. patent 5,918,223, June 1999.

R. Bod. Memory-based models of melodic analysis: Challenging the Gestalt principles. *Journal of New Music Research*, 31(1):27–37, 2001.

J. Bonada. Automatic technique in frequency domain for near-lossless time-scale modification of audio. In *Proceedings International Computer Music Conference*, pages 396–399, 2000.

J. Bonada and X. Serra. Synthesis of the singing voice by performance sampling and spectral models. *IEEE Signal Processing Magazine*, 24(2):67–79, 2007.

J. Bonada, M. Blaauw, À. Loscos, and H. Kenmochi. Unisong: A choir singing synthesizer. In *Proceedings of 121st Convention of the AES*, 2006.

A. Bregman. Psychological data and computational auditory scene analysis. In D. Rosenthal and H. Okuno, editors, *Computational auditory scene analysis*. Lawrence Erlbaum Associates Inc., 1998.

A. Bregman. *Auditory scene analysis*. MIT Press, Harvard, MA, 1990.

G. Brown. *Computational auditory scene analysis: A representational approach*. Unpublished PhD thesis, University of Sheffield, 1992.

J. Brown. Determination of the meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94(4):1953–1957, 1993.

J. Brown and M. Puckette. An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

D. Byrd. *Music notation by computer*. Unpublished PhD thesis, Indiana University, 1984.

E. Cambouropoulos. The local boundary detection model and its application in the study of expressive timing. In *Proceedings International Computer Music Conference*, 2001.

P. Cano. Fundamental frequency estimation in the SMS analysis. In *Proceedings Digital Audio Effects Conference*, 1998.

P. Cano. *Content-based audio search: From fingerprinting to semantic audio retrieval*. Unpublished PhD thesis, University Pompeu Fabra, Barcelona, 2007.

P. Cano, À. Loscos, J. Bonada, M. de Boer, and X. Serra. Voice morphing system for impersonating in karaoke applications. In *Proceedings International Computer Music Conference*, 2000.

P. Cano, E. Batlle, H. Mayer, and H. Neuschmied. Robust sound modeling for song detection in broadcast audio. In *Proceedings AES 112th International Convention*, 2002.

P. Cano, M. Koppenberger, S. Ferradans, A. Martinez, F. Gouyon, V. Sandvold, V. Tarasov, and N. Wack. MTG-DB: A repository for music audio processing. In *Proceedings International Conference on Web Delivering of Music*, 2004.

P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, 41(3):271–284, 2005a.

P. Cano, O. Celma, M. Koppenberger, and J. Martin-Buldú. The topology of music artists' graphs. In *Proceedings XII Congreso de Fisica Estadistica*, 2005b.

P. Cano, M. Koppenberger, N. Wack, J. Garcia, J. Masip, O. Celma, D. Garcia, E. Gómez, F. Gouyon, E. Guaus, P. Herrera, J. Massaguer, B. Ong, M. Ramirez, S. Streich, and X. Serra. An industrial-strength content-based music recommendation system. In *Proceedings International ACM SIGIR Conference*, page 673, 2005c.

P. Cano, O. Celma, M. Koppenberger, and J. Martin-Buldú. Topology of music recommendation networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 16, 2006a.

P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack. ISMIR 2004 audio description contest. *MTG Technical Report MTG-TR-2006-02*, 2006b.

M. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *Proceedings International Computer Music Conference*, 2000.

O. Celma. Foafing the music: Bridging the semantic gap in music recommendation. In *Proceedings 5th International Semantic Web Conference*, 2006a.

O. Celma. *Music Recommendation: a multi-faceted approach*. Unpublished DEA thesis, University Pompeu Fabra, 2006b.

A. Cemgil and B. Kappen. Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81, 2003.

A. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28(4):259–273, 2001.

D. Childers. Speech processing and synthesis for assessing vocal disorders. *IEEE Magazine on Engineering in Medicine and Biology*, 9:69–71, 1990.

L. Cohen. Time-frequency distributions - A review. *Processings of the IEEE*, 77 (7):941–981, 1989.

N. Collins. Using a pitch detector for onset detection. In *International Conference on Music Information Retrieval*, pages 100–106, 2005.

G. Cooper and L. Meyer. *The rhythmic structure of music*. University of Chicago Press, 1960.

K. de Koning and S. Oates. Sound base: Phonetic searching in sound archives. In *Proceedings International Computer Music Conference*, pages 433–466, 1991.

P. Desain and H. Honing. The quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):55–66, 1989.

P. Desain and H. Honing. Computational models of beat induction: The rule-based approach. *Journal of New Music Research*, 28(1):29–42, 1999.

S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.

S. Dixon and E. Cambouropoulos. Beat tracking with musical knowledge. In *Proceedings European Conference on Artificial Intelligence*, pages 626–630, 2000.

S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In *Proceedings International Conference on Music Information Retrieval*, pages 159–165, 2003.

S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings International Conference on Music Information Retrieval*, pages 509–516, 2004.

S. Dixon, W. Goebl, and E. Cambouropoulos. Perceptual smoothness of tempo in expressively performed music. *Music Perception*, 23(3):195–214, 2006.

J. Downie, editor. *The MIR/MDL evaluation project white paper collection - Proceedings International Conference on Music Information Retrieval*. 2nd edition, 2002.

J. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37:295–340, 2003a.

J. Downie. The scientific evaluation of music information retrieval systems: Foundations and the future. *Computer Music Journal*, 28(2):12–23, 2003b.

J. Downie. The MusiFind musical information retrieval project, phase II: User assessment survey. In *Proceedings 22nd Annual Conference of the Canadian Association for Information Science*, pages 149–166, 1994.

C. Drake, L. Gros, and A. Penel. How fast is that music? The relation between physical and perceived tempo. In S. Yi, editor, *Music, Mind and Science*, pages 190–203. Seoul National University Press, 1999.

B. Eaglestone. A database environment for musician-machine interaction experimentation. In *Proceedings International Computer Music Conference*, pages 20–27, 1988.

B. Eaglestone and A. Verschoor. An intelligent music repository. In *Proceedings International Computer Music Conference*, pages 437–440, 1991.

D. Ellis. *Prediction-driven computational auditory scene analysis*. Unpublished PhD thesis, Massachussetts Institute of Technology, 1996.

L. Fabig and J. Janer. Transforming singing voice expression - The sweetness effect. In *Proceedings Digital Audio Effects Conference*, pages 70–75, 2004.

B. Feiten, R. Frank, and T. Ungvary. Organizing sounds with neural nets. In *Proceedings International Computer Music Conference*, pages 441–444, 1991.

J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings IEEE International Conference on Multimedia and Expo*, pages 452–455, 2000.

J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *Proceedings International Conference on Music Information Retrieval*, pages 265–266, 2002.

T. Fujishima. Real-time chord recognition of musical sound: A system using common lisp music. In *International Computer Music Conference*, pages 464–467, 1999.

D. Godsmark and G. J. Brown. A blackboard architecture for computational auditory scene analysis. *Speech Communication*, 27:351–366, 1999.

B. Gold and L. Rabiner. Parallel processing techniques for estimating pitch periods of speech in the time domain. *Journal of the Acoustical Society of America*, 46:442–448, 1969.

L. Gomes, P. Cano, E. Gómez, M. Bonnet, and E. Batlle. Audio watermarking and fingerprinting: For which applications? *Journal of New Music Research*, 32(1):65–82, 2003.

E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Issue on Computation in Music*, 18(3): 294–301, 2004.

E. Gómez. *Tonal Description of Music Audio Signals*. Unpublished PhD thesis, University Pompeu Fabra, Barcelona, 2006.

E. Gómez, F. Gouyon, P. Herrera, and X. Amatriain. MPEG-7 for content-based music processing. In *Proceedings 4th WIAMIS-Special session on Audio Segmentation and Digital Music*, 2003a.

E. Gómez, F. Gouyon, P. Herrera, and X. Amatriain. Using and enhancing the current MPEG-7 standard for a music content processing tool. In *Proceedings 114th AES Convention*, 2003b.

E. Gómez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32 (1):23–40, 2003c.

E. Gómez, G. Peterschmitt, X. Amatriain, and P. Herrera. Content-based melodic transformations of audio for a music processing application. In *Proceedings Digital Audio Effects Conference*, 2003d.

E. Gómez, J. P. Bello, M. Davies, D. Garcia, F. Gouyon, C. Harte, P. Herrera, C. Landone, K. Noland, B. Ong, V. Sandvold, S. Streich, and B. Wang. Front-end signal processing and low-level descriptors computation module. Technical Report D2.1.1, SIMAC IST Project, 2005.

J. Gordon. *Perception of attack transients in musical tones*. Unpublished PhD thesis, CCRMA, Stanford University, 1984.

M. Goto. A robust predominant-f0 estimation method for real-time detection of melody and bass lines in CD recordings. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 757–760, 2000.

M. Goto. An audio-based real-time beat tracking system for music with or without drums. *Journal of New Music Research*, 30(2):159–171, 2001.

M. Goto and Y. Muraoka. Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, (27): 311–335, 1999.

F. Gouyon. *A computational approach to rhythm description*. Unpublished PhD thesis, Pompeu Fabra University, Barcelona, 2005.

F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

F. Gouyon and P. Herrera. Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors. In *Proceedings Audio Engineering Society, 114th Convention*, 2003.

F. Gouyon and B. Meudic. Towards rhythmic content processing of musical signals: Fostering complementary approaches. *Journal of New Music Research*, 32(1):41–64, 2003.

F. Gouyon, L. Fabig, and J. Bonada. Rhythmic expressiveness transformations of audio recordings: Swing modifications. In *Proceedings Digital Audio Effects Conference*, pages 94–99, 2003a.

F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings 25th International AES Conference*, pages 196–204, 2004.

F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Speech and Audio Processing*, 14:1832–1844, 2006a.

F. Gouyon, G. Widmer, X. Serra, and A. Flexer. Acoustic cues to beat induction: A Machine Learning perspective. *Music Perception*, 24(2):181–194, 2006b.

M. Grachten, J. Arcos, and R. López de Mántaras. A case based approach to expressivity-aware tempo transformation. *Machine Learning Journal*, 65(2-3): 411–437, 2006.

I. Guyon, S. Gunn, A. Ben Hur, and G. Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Proceedings Neural Information Processing Systems Conference*, pages 545–552, 2004.

S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15:2385–2395, 2004.

H. Harb and L. Chen. Robust speech music discrimination using spectrum's first order statistics and neural networks. In *Proceedings 7th International Symposium on Signal Processing and Its Applications*, pages 125–128, 2003.

M. Hearst. User interfaces and visualization. In R. Baeza-Yates and B. Ribeiro-Neto, editors, *Modern information retrieval*. Harlow, Essex: ACM Press, 1999.

P. Herrera. *Automatic classification of percussion sounds: From acoustic features to semantic descriptions*. Unpublished PhD thesis, University Pompeu Fabra, in print.

P. Herrera and J. Bonada. Vibrato extraction and parameterization in the spectral modeling synthesis framework. In *Proceedings Digital Audio Effects Conference*, 1998.

W. Hess. *Pitch Determination of Speech Signals. Algorithms and Devices.* Springer Series in Information Sciences. Springer-Verlag, Berlin, New York, Tokyo, 1983.

S. Hoek. Method and apparatus for signal processing for time-scale and/or pitch modification of audio signals, U.S. patent 6266003, 1999.

H. Honing. From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3):50–61, 2001.

J. Janer, J. Bonada, and S. Jordà. Groovator - An implementation of real-time rhythm transformations. In *Proceedings 121st Convention of the Audio Engineering Society*, 2006.

T. Jehan. *Musical signal parameter estimation*. Unpublished MSc thesis, Institut de Formation Supérieure en Informatique et Communication, Université Rennes I, 1997.

K. Jenssen. Envelope model of isolated musical sounds. In *Proceedings Digital Audio Effects Conference*, 1999.

I. Jermyn, C. Shaffrey, and N. Kingsbury. The methodology and practice of the evaluation of image retrieval systems and segmentation methods. Technical Report 4761, Institut National de la Recherche en Informatique et en Automatique, 2003.

K. Kashino and H. Murase. Sound source identification for ensemble music based on the music stream extraction. In *Proceedings International Joint Conference on Artificial Intelligence Workshop of Computational Auditory Scene Analysis*, pages 127–134, 1997.

K. Kashino, T. Kinoshita, and H. Tanaka. Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In *Proceedings International Joint Conference On Artificial Intelligence*, 1995.

M. Kassler. Toward musical information retrieval. *Perspectives of New Music*, 4:59–67, 1966.

D. Keislar, T. Blum, J. Wheaton, and E. Wold. Audio analysis for content-based retrieval. In *Proceedings International Computer Music Conference*, pages 199–202, 1995.

D. Kirovski and H. Attias. Beat-ID: Identifying music via beat analysis. In *Proceedings IEEE International Workshop on Multimedia Signal Processing*, pages 190– 193, 2002.

A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3089 – 3092, 1999.

A. Klapuri. *Signal Processing Methods for the Automatic Transcription of Music*. Unpublished PhD thesis, Tampere University of Technology, Tampere, Finland, 2004.

A. Klapuri and M. Davy, editors. *Signal processing methods for music transcription*. Springer-Verlag, New York, 2006.

A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Trans. Speech and Audio Processing*, 14(1):342–355, 2006.

B. Kostek. Computer-based recognition of musica phrases using the rough-set approach. *Information Sciences*, 104:15–30, 1998.

M. Kotti, L. Martins, E. Benetos, J. Santos Cardoso, and C. Kotropoulos. Automatic speaker segmentation using multiple features and distance measures: A comparison of three approaches. In *Proceedings IEEE International Conference on Multimedia and Expo*, pages 1101–1104, 2006.

R. Kronland-Martinet, J. Morlet, and Grossman. Analysis of sound patterns through wavelet transforms. *International Journal on Pattern Recognition and Artificial Intelligence*, 1(2):273–302, 1987.

C. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York, 1990.

E. Lapidaki. *Consistency of tempo judgments as a measure of time experience in music listening*. Unpublished PhD thesis, Northwestern University, Evanston, IL, 1996.

E. Large and E. Kolen. Resonance and the perception of musical meter. *Connection Science*, 6:177–208, 1994.

J. Laroche. Traitement des signaux audio-fréquences. Technical report, Ecole National Supérieure de Télécommunications, 1995.

J. Laroche. Estimating tempo, swing and beat locations in audio recordings. In *Proceedings IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 135–138, 2001.

J. Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233, 2003.

J. Laroche and M. Dolson. Improved phase-vocoder. time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7:323–332, 1999.

S. Lee, H. D. Kin, and H. S. Kim. Variable time-scale modification of speech using transient information. In *Proceedings International Conference of Acoustics, Speech, and Signal Processing*, volume 2, pages 1319 – 1322, 1997.

M. Leman. Foundations of musicology as content processing science. *Journal of Music and Meaning*, 1(3), 2003. URL `http://www.musicandmeaning.net/index.php`.

F. Lerdahl and R. Jackendoff. *A generative theory of tonal music*. MIT Press, Cambridge, Massachusetts, 1983.

M. Lesaffre, M. Leman, K. Tanghe, B. De Baets, H. De Meyer, and J. Martens. User-dependent taxonomy of musical features as a conceptual framework for musical audio-mining technology. In *Proceedings Stockholm Music Acoustics Conference*, 2003.

M. Lew, N. Sebe, and J. Eakins. Challenges of image and video retrieval. In *Proceedings International Conference on Image and Video Retrieval*, pages 1–6, 2002.

H. Lincoln. Some criteria and techniques for developing computerized thematic indices. In H. Heckman, editor, *Electronishe Datenverarbeitung in der Musikwissenschaft*. Regensburg: Gustave Bosse Verlag, 1967.

À. Loscos. *Spectral Processing of the Singing Voice*. Unpublished PhD thesis, University Pompeu Fabra, Barcelona, 2007.

À. Loscos and J. Bonada. Emulating rough and growl voice in spectral domain. In *Proceedings Digital Audio Effects Conference*, 2004.

E. Maestre and E. Gómez. Automatic characterization of dynamics and articulation of expressive monophonic recordings. In *Proceedings 118th Audio Engineering Society Convention*, 2005.

R. Maher and J. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 95:2254–2263, 1993.

B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Language*. Wiley and Sons, New York, 2002.

D. Marr. *Vision*. W.H. Freeman and Co., San Fransisco, 1982.

S. McAdams. Audition: Physiologie, perception et cognition. In J. Requin, M. Robert, and M. Richelle, editors, *Traite de psychologie expérimentale*, pages 283–344. Presses Universitaires de France, 1994.

S. McAdams and E. Bigand. *Thinking in Sound: The Cognitive Psychology of Human Audition*. Clarendon, Oxford, 1993.

R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744–754, 1986.

R. McNab, L. Smith, and I. Witten. Signal processing for melody transcription. In *Proceedings 19th Australasian Computer Science Conference*, 1996.

G. McNally. Variable speed replay of digital audio with constant output sampling rate. In *Proceedings 76th AES Convention*, 1984.

R. Meddis and M. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. *Journal of the Acoustical Society of America*, 89(6):2866–2882, 1991.

D. Mellinger. *Event formation and separation in musical sound*. Unpublished PhD thesis, Stanford University, 1991.

M. Melucci and N. Orio. Musical information retrieval using melodic surface. In *Proceedings ACM Conference on Digital Libraries*, pages 152–160, 1999.

A. Meng. *Temporal Feature Integration for Music Organisation*. Unpublished PhD thesis, Technical University of Denmark, 2006.

M. Minsky and O. Laske. A conversation with Marvin Minsky. *AI Magazine*, 13(3):31–45, 1992.

B. Moore. *Hearing – Handbook of perception and cognition*. Academic Press Inc., London, 2nd edition, 1995.

E. Moulines, F. Charpentier, and C. Hamon. A diphone synthesis system based on time-domain prosodic modifications of speech. In *Proceedings International Conference of Acoustics, Speech, and Signal Processing*, volume 1, pages 238–241, 1989.

N. Nettheim. On the spectral analysis of melody. *Journal of New Music Research*, 21:135–148, 1992.

A. Noll. Cepstrum pitch determination. *Journal of the Acoustical Society of America*, 41:293–309, 1967.

B. Ong. *Structural Analysis and Segmentation of Music Signals*. Unpublished PhD thesis, University Pompeu Fabra, Barcelona, 2007.

A. Oppenheim and R. Schafer. From frequency to quefrency: A history of the cepstrum. *IEEE Signal Processing Magazine*, 21(5):95–106, 2004.

N. Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90, 2006.

F. Pachet and P. Roy. Reifying chords in automatic harmonization. In *Proceedings ECAI Workshop on Constraints for Artistic Applications*, 1998.

F. Pachet, P. Roy, and D. Cazaly. A combinatorial approach to content-based music selection. *IEEE Multimedia*, 7:44–51, 2000.

N. Pal and S. Pal. A review of image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

E. Pampalk. *Computational Models of Music Similarity and their Application to Music Information Retrieval*. Unpublished PhD thesis, Vienna University of Technology, Vienna, 2006.

E. Pampalk and M. Gasser. An implementation of a simple playlist generator based on audio similarity measures and user feedback. In *Proceedings International Conference on Music Information Retrieval*, pages 389–390, 2006.

E. Pampalk and M. Goto. MusicRainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proceedings International Conference on Music Information Retrieval*, pages 367–370, 2006.

R. Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11(4):409–464, 1994.

D. Pearce and H. Hirsch. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings International Conference on Spoken Language Processing*, 2000.

G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, CUIDADO IST Project, 2004.

G. Peeters, A. La Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *International Conference on Music Information Retrieval*, pages 94–100, 2002.

J. Pinquier, J.-L. Rouas, and R. André-Obrecht. A fusion study in speech/music classification. In *Proceedings International Conference on Multimedia and Expo*, pages 409–412, 2003.

M. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24: 243–248, 1976.

M. Przybocki and A. Martin. NIST speaker recognition evaluations. In *Proceedings International Conference on Language Resources and Evaluations*, pages 331–335, 1989.

H. Purwins. *Profiles of Pitch Classes Circularity of Relative Pitch and Key – Experiments, Models, Computational Music Analysis, and Perspectives*. Unpublished PhD thesis, Technical University of Berlin, 2005.

H. Purwins, B. Blankertz, and K. Obermayer. A new method for tracking modulations in tonal music in audio data format. *Proceeding International Joint Conference on Neural Network*, pages 270–275, 2000.

H. Purwins, T. Graepel, B. Blankertz, and K. Obermayer. Correspondence analysis for visualizing interplay of pitch class, key, and composer. In E. Puebla, G. Mazzola, and T. Noll, editors, *Perspectives in Mathematical Music Theory*. Verlag, 2003.

L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings IEEE*, 77(2):257–285, 1989.

L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

L. Rabiner, M. Sambur, and C. Schmidt. Applications of a nonlinear smoothing algorithm to speech processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(6):552–557, 1975.

R. Ramirez and A. Hazan. A tool for generating and explaining expressive music performances of monophonic jazz melodies. *International Journal on Artificial Intelligence Tools*, 15(4):673–691, 2006.

R. Ramirez, A. Hazan, E. Gómez, and E. Maestre. A machine learning approach to expressive performance in jazz standards. In *Proceedings International Conference on Knowledge Discovery and Data Mining*, 2004.

C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

C. Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137(1-2):217–238, 2002.

E. Ravelli, J. Bello, and M. Sandler. Automatic rhythm modification of drum loops. *IEEE Signal Processing Letters*, 14(4):228–231, 2007.

S. Rossignol. *Séparation, segmentation et identification d'objets sonores. Application á la représentation, á la manipulation des signaux sonores, et au codage dans les applications multimedias*. Unpublished PhD thesis, IRCAM, Paris, France, 2000.

S. Rossignol, X. Rodet, P. Depalle, J. Soumagne, and J.-L. Collette. Vibrato: Detection, estimation, extraction, modification. In *Proceedings Digital Audio Effects Conference*, 1999.

K. Sakakibara, L. Fuks, H. Imagawa, and N. Tayama. Growl voice in ethnic and pop styles. In *Proceedings International Symposium on Musical Acoustics*, 2004.

E. Scheirer. *Music listening systems*. Unpublished PhD thesis, Massachusets Institute of Technology, 2000.

E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proceedings IEEE International Conference on Audio, Speech and Signal Processing*, pages 1331–1334, 1997.

J. Schoentgen. Stochastic models of jitter. *Journal of the Acoustical Society of America*, 109:1631–1650, 2001.

E. Selfridge-Field. Conceptual and representational issues in melodic comparison. In W. B. Hewlett and E. Selfridge-Field, editors, *Melodic Similarity: Concepts, Procedures, and Applications*. MIT Press, Cambridge, Massachusetts, 1998.

X. Serra. *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*. Unpublished PhD thesis, Stanford University, 1989.

X. Serra. Sound hybridization techniques based on a deterministic plus stochastic decomposition model. In *Proceedings International Computer Music Conference*, 1994.

X. Serra and J. Bonada. Sound transformations based on the SMS high level attributes. In *Proceedings Digital Audio Effects Conference*, Barcelona, 1998.

A. Sheh and D. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proceedings International Conference on Music Information Retrieval*, 2003.

M. Slaney, M. Covell, and B. Lassiter. Automatic audio morphing. In *Proceedings IEEE International Conference on Audio, Speech and Signal Processing*, pages 1001–1004, 1996.

P. Smaragdis. *Redundancy Reduction for Computational Audition, a Unifying Approach*. Unpublished PhD thesis, Massachusetts Institute of Technology, 2001.

A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.

J. Smith and X. Serra. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings Internationtal Computer Music Conference*, pages 290–297, 1987.

C. Spevak, B. Thom, and K. Hothker. Evaluating melodic segmentation. In *Proceedings International Conference on Music and Artificial Intelligence*, 2002.

J. Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987.

E. Terhardt, G. Stoll, and M. Seewann. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *Journal of the Acoustical Society of America*, 71:679–688, 1981.

H. Thornburg and F. Gouyon. A flexible analysis/synthesis method for transients. In *Proceedings International Computer Music Conference*, 2000.

P. Toiviainen and T. Eerola. A method for comparative analysis of folk music based on musical feature extraction and neural networks. In *Proceedings*

International Symposium on Systematic and Comparative Musicology, and International Conference on Cognitive Musicology, 2001.

M. Towsey, A. Brown, S. Wright, and J. Diederich. Towards melodic extension using genetic algorithms. *Educational Technology & Society*, 4(2), 2001.

R. Typke. *Music Retrieval based on Melodic Similarity*. PhD thesis, Utrecht University, 2007.

G. Tzanetakis. *Manipulation, analysis and retrieval systems for audio signals*. Unpublished PhD thesis, Computer Science Department, Princeton University, June 2002.

G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *Proceedings IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 103–106, 1999.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

A. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In *Proceedings ACM Conference on Multimedia*, pages 235–240, 1998.

V. Verfaille, U. Zölzer, and D. Arfib. Adaptive digital audio effects (A-DAFx): A new class of sound transformations. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1817– 1831, 2006.

W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *Proceedings IEEE International Conference of Acoustics, Speech, and Signal Processing*, volume 2, pages 554–557, 1993.

T. Verma, S. Levine, and T. Meng. Transient modeling synthesis: A flexible analysis/synthesis tool for transient signals. In *Proceedings Internationtal Computer Music Conference*, 1997.

E. Vidal and A. Marzal. A review and new approaches for automatic segmentation of speech signals. In L. Torres, E. Masgrau, and Lagunas M., editors, *Signal Processing V: Theories and Applications*. 1990.

P. Walmsley, S. Godsill, and P. Rayner. Bayesian graphical models for polyphonic pitch tracking. In *Proceedings Diderot Forum*, 1999.

D. Wang and G. Brown, editors. *Computational auditory scene analysis – Principles, Algorithms, and Applications*. IEEE Press - Wiley-Interscience, 2006.

J. Wayman, R. Reinke, and D. Wilson. High quality speech expansion, compression, and noise filtering using the SOLA method of time scale modification. In *Proceedings 23d Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 714–717, 1989.

B. Whitman. *Learning the Meaning of Music*. Unpublished PhD thesis, Massachusetts Institute of Technology, MA, USA, 2005.

M. Wright and E. Berdahl. Towards machine learning of expressive microtiming in Brazilian drumming. In *Proceedings International Computer Music Conference*, 2006.

U. Zölzer, editor. *DAFX Digital Audio Effects*. Wiley & Sons, 2002.