2

AUDIO DESCRIPTION

Music information retrieval systems employ a multitude of techniques for the computational description of musical audio content. This chapter provides an introduction to this vast topic, by reviewing some of the most important audio descriptors (section 2.1) and a number of applications of these descriptors that are relevant to the rest of the work in this thesis (section 2.2).

2.1 AUDIO FEATURES

One conceptual framework that is typically used to write about and reflect on music descriptors distinguishes between low-level and high-level descriptions. Low-level features describe the properties of an audio sample on the level of the signal. High-level features correspond to the abstractions found in musical scores and natural language [179]. While low-level features tend to suit the language of machines and mathematics, high-level features are the ones that are used by humans (users of a music app, musicians, music scholars). And while machines are excellently equipped to make accurate measurements over a signal, humans, on the other hand, discuss and reason about music using a personal and highly 'enculturated' set of abstractions that changes over time and varies from individual to individual [5]. The discrepancy between signal-level and semantic-level music descriptions is complex: low-level descriptions may refer to not just signal-level, but also physical and sensory attributes of sound, and high-level representations can relate to formal, cognitive, or social aspects of it. The notion of mid-level features or descriptors is

sometimes also used to refer to an intermediate class of representations, e.g. in [107], where it is equated, roughly, to a level that aims to approximate perception.

The computational modeling challenges associated with this discrepancy are sometimes referred to as the 'semantic gap'. It has proven very challenging to teach a computer about these 'semantic' aspects of a piece of music. Modeling such high-level representations typically involves a trained classifier or probabilistic model learned from annotated data. Even advanced models, however, may not always yield reliable representations, and high-level representations often exhibit a significant trade-off between usefulness and reliability [179]. Questions have also been raised as to whether 'semantics', the elusive, high-level information some see as a holy grail of information retrieval, are present at all, in the audio representations used. The notions of 'semantics' and the 'semantic gap' may therefore be illusory, as Wiggins suggests in [202]. Nonetheless, the ambition to improve low- mid- and high-level representations to address increasingly high-level search problems continues to be a primary concern in music information retrieval.

The efforts put in by the audio content-based retrieval community have spawned an impressive collection of descriptors, low-, mid- and high-level. Many of these descriptions relate to one of the main 'dimensions' or parameters of music traditionally recognized in music theory: melody, harmony, rhythm and timbre. We will list a few of the most used and most relevant descriptors, with some explanation of how and why they may be used, in sections 2.1.1–2.1.4.

Features that do not directly fall into any of the categories along the low-level – high-level axis as introduced above include psycho-acoustic features and learned features. Psycho-acoustic features measure a specific psycho-acoustic attribute of a sound. They are often based on low-level signal measurements, but could be seen as high-level in that they approximate a human rating on a scale. Psycho-acoustic features are further discussed in section 2.1.5.

Learned features have come up more recently, out of the work done on feature learning for music content description. In feature learn-

ing applications, a content-based music description system might perform a classification of low-level music representations into high-level categories like genre, mood or a latent factor in a set of user behavior data, using advanced machine learning techniques. The classifier then yields, as an intermediate step in its pipeline, one or more novel, learned feature representations that are better suited to accomplish the original task (on the same or a larger dataset), or another one. Feature learning is discussed further in section 2.1.6.

2.1.1 *Basis Features*

The Fourier Transform

Many of the audio descriptors in this chapter are based on frequency information. To compute the amplitude of a signal at specific frequencies from an audio signal, the discrete signal $y(n)$ is converted to its frequency representation $Y(k)$ using the Fourier transform:

$$Y(k) = \sum_{n=0}^{N-1} y(n)\, e^{-2\pi i k n/N}, \quad k = 0 \ldots N-1 \tag{1}$$

yielding a single complex-valued spectrum for the entire time series $y(n)$.

Typically, the complex-valued $Y(k)$ will be represented in spherical coordinates, with *magnitude* $|Y(k)|$ and *phase* angle denoted $\phi(k)$:

$$Y(k) = |Y(k)|\, e^{-i\phi(k)} \iff \begin{array}{ll} |Y(k)| & \\ \phi(k) & = -i \ln(\frac{Y(k)}{|Y(k)|}). \end{array} \tag{2}$$

In many cases, only the magnitude of the Fourier transform is used.

To be able to look at the evolution of the frequency content, the Fourier transform is typically computed for an array of short overlapping windows in the signal, in a procedure called the *short term Fourier transform* (STFT). The result is a time series of windowed spectra, together forming the *spectrogram*.

$$Y(j,k) = \sum_{n=0}^{M-1} w(n)\, y(n + jH)\, e^{-2\pi i k n/N} \qquad \begin{array}{l} j = 0 \ldots \lfloor M/N \rfloor \\ k = 0 \ldots M-1 \end{array} \tag{3}$$

where M is the total length of the time series and N the length of the window w that is applied.

The frequencies to which k corresponds depend on the length of the window N and the sample rate of the original audio f_s (e.g., 44100 Hz):

$$f = \frac{k}{N} f_s \tag{4}$$

Furthermore, the Fourier transform of a real-valued signal $y(n)$ always yields a complex-valued spectrum $Y(k)$ for which the magnitudes $|Y(k)|$ are symmetric around $N/2$, and the phases are *antisymmetric* around $N/2$. As a result, all frequency information is contained in $k = 1 \ldots N/2$. The frequency f corresponding to $k = N/2$ is called $f_N = f_s/2$, the *Nyquist frequency*.

From this point on, we will abandon the standard notation for the discrete frequency analysis used above, in favor of the continuous conventions, i.e., we will express frequency representations in terms of f, rather than k, time as t instead of n. We will also assume frequency axes go up to f_N rather than f_s. Finally, throughout all of this thesis, we will keep using arguments for indexing; i.e., we write $Y(j,k)$ rather than $Y_{j,k}$; it will make formulas more readable.

Frequency Scales

The linear frequency scale f of the Fourier transform has some clear advantages. For example, a harmonic series of pure sinusoids will be represented as a series of equidistant peaks along the frequency axis, aiding a number of computations such as estimating the fundamental frequency of a complex tone. At other times however, a logarithmic division of the frequency axis may be more appropriate. Human perception of pitch follows *Weber's Law*. This law states that the smallest noticeable difference in some perceived quantity is, roughly, a constant percentage of the quantity in question, indicating a logarithmic sensitivity to the stimulus [111]. Similarly, the Western musical pitch scale follows a logarithmic function of frequency.

The constant-Q transform is similar to the Fourier transform, but follows a logarithmic division of the frequency axis [19]. Its name

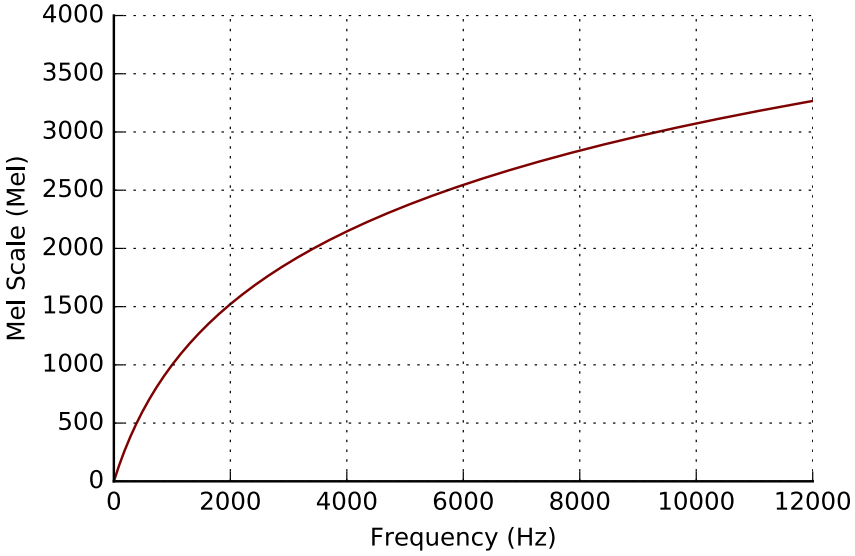

Figure 2.: The mel scale as a function of linear frequency.

refers to the constant relative width Q of the filters that would be used if the transform were to be implemented using a filter bank. The constant-Q transform and the constant-Q spectrogram are excellent basis features for pitch and harmony description and will be used as such in section 2.1.3. Some of the practical challenges in computing a constant-Q transform with useful resolutions are discussed in [137].

Experiments with human listeners have shown that, for a more accurate model of human pitch height judgements, Weber's law must be refined. Section 2.1.5 describes this in more detail, but in brief, the general observation is that the inner ear's representation of pitch is part linear with frequency, part logarithmic. One frequency scale that incorporates this is the mel scale (figure 2). It is roughly linear below 1000 Hz and logarithmic above 1000 Hz [111], but has no one formula. A commonly used formula is:

$$m(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{5}$$

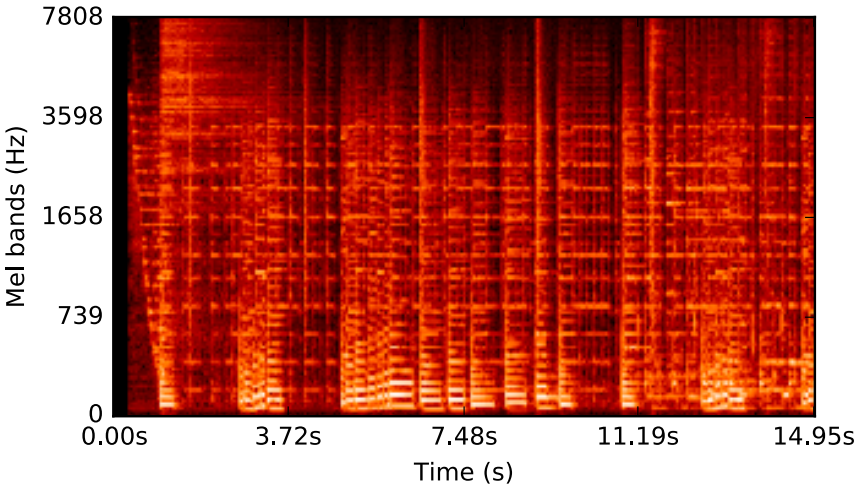

Figure 3.: Mel spectrogram. The song fragment shown begins with a piano playing a downward glissando.

This scale is the basis for much of the early work done in speech recognition, and a widely used basis feature for timbre description. The mel scale can be used to compute mel spectrograms, similar to the STFT-based linear spectrogram, but with a different frequency axis, as in figure 3.

2.1.2 *Timbre Description*

Timbre is a complex attribute of sound that is not easily defined. Timbre pertains to the 'tone color' or texture of a sound when pitch and loudness remain the same, and is crucial in our ability to recognize the differences between different instruments [30].

Timbre features, and particularly low-level timbre features, constitute a large number of the audio descriptors typically encountered in the music information retrieval literature. This may in part be explained by their success at predicting a particular notion of music similarity, mood, and musical genre. Some of these descriptors are

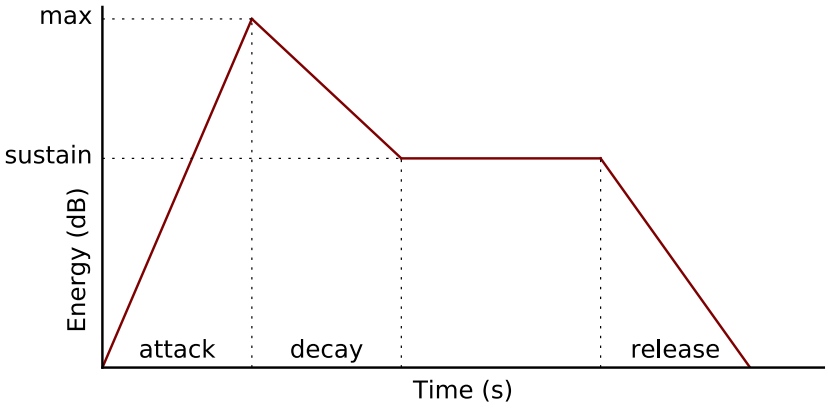

Figure 4.: Schematic of the attack, decay, sustain and release parameters of the temporal envelope associated with a note event.

now introduced, beginning with features that summarize the shape of the temporal and spectral envelopes of a sound.

Temporal Domain

In the temporal domain, the temporal envelope of a single note event is typically characterized by its attack time (the time it takes for its amplitude to reach an initial peak), its decay time after the initial amplitude peak, the sustain amplitude that is maintained until release of the note, and its release time after this point. Attack, decay, sustain and release are illustrated in figure 4. These descriptors are mostly useful for the description of single events, e.g. in the classification of instrument samples [154]. Another often-recurring time-domain is the zero crossing rate, the number of times the sign of an audio signal changes over a specified time window.

Frequency Domain

In the frequency domain, the spectral envelope is often parametrized by its first statistical moments, as if the spectrum were a statistical

distribution: the spectral centroid (the distribution mean), spectral spread (the distribution variance):

$$\text{centroid} = \sum_f f\, Y'(f) \tag{6}$$

$$\text{spread} = \sum_f (f - \text{centroid})^2\, Y'(f) \tag{7}$$

Spectral skewness and spectral kurtosis can be used too. These are the straightforward extensions of the centroid and the variance, again considering the amplitude spectrum as distribution:

$$\text{skewness} = \sum_f \frac{(f - \text{centroid})^3\, Y'(f)}{\text{spread}^3} \tag{8}$$

$$\text{kurtosis} = \sum_f \frac{(f - \text{centroid})^4\, Y'(f)}{\text{spread}^4} \tag{9}$$

where $Y'(f)$ is the magnitude spectrum, but normalized to sum to 1 [154]. Other spectral shape descriptors include the spectral roll-off point, marking the frequency below which 95% of the spectral energy is contained.

A very widely-used set of timbre descriptor are the MFCC features. Originating from the speech processing domain, mel frequency cepstrum coefficients describe the shape of the mel scale spectral envelope, by breaking it down into maximally de-correlated components: cosine-shaped basis functions referred to as *cepstral* components. Concretely: a mel scale amplitude spectrum $Y_m(m)$ is computed, after which the logarithm of the amplitudes is taken, and the coefficients of the components are obtained using a discrete cosine transformation (DCT) on the resulting envelope [127]:

$$\text{MFCC} = \text{DCT}(\log(Y_m(m))). \tag{10}$$

The DCT is a linear transformation in which a vector is expressed as a sum of cosines of different frequency $\{0, 2N, N, N/2, N/3, \ldots\}$. The coefficients of the summed cosines form the MFCC. Usually around 12 or 13 are used, of which the first is proportional to the sum of

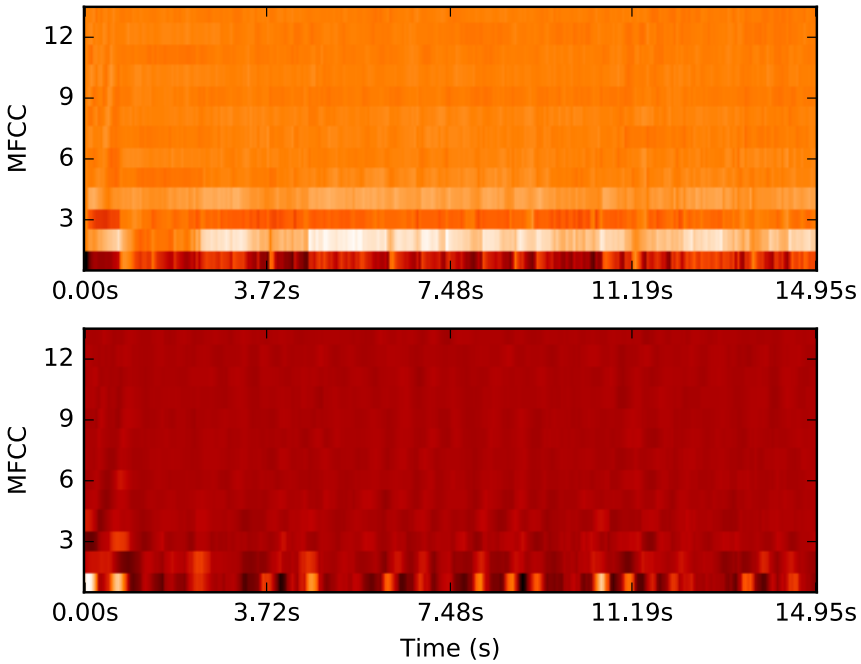

Figure 5.: MFCC and ΔMFCC features for a series of audio frames.

all amplitude bins, and therefore highly correlated to the spectral energy. When MFCC's are computed over a series of frames, it may be useful to compute so-called ΔMFCC and Δ^2MFCC features, the frame-wise differences of the MFCC and ΔMFCC, respectively. MFCC and ΔMFCC are shown in figure 5.

MFCC features have been used successfully as a basis for classification in a variety of tasks. As a result they have become the most used feature for the frame-based spectral (envelope) description, despite having been developed for non-musical applications first, and having seen some of its perceptual justifications refuted [6, 137].

Timbre features in general are widely used throughout MIR, perhaps because they have shown to work for a variety of tasks, perhaps because, as mostly low-level features, they are typically easy to compute and understand from an acoustics or signal processing perspec-

tive, whereas harmony and melody descriptors are more often rooted in music theory or music perception.

2.1.3 *Harmony Description*

Harmony, by definition, involves the simultaneous sounding of two or more pitches. Harmony description therefore relies on an accurate estimation of the pitch content in an audio segment. While the constant-Q transform provides a reasonable interface to this information, one particular alternative has turned out to be more practical: chroma features. We first introduce the notion of *pitch class* and the *pitch helix*, then discuss chroma.

Mathematical models of pitch

Both music theory and music perception describe a notion of *octave equivalence*. If a pitch is one octave above another, i.e., its fundamental frequency is two times the other's, the two are perceived as very similar. This effect is transitive: any two pitches that are spaced an integer number of octaves apart, will be perceived as similar. In the context of the standard equal-tempered scale, this establishes an equivalence relationship between each of the 12 pitches in an octave, and all of the pitches n octaves above and below it ($n \in \mathbb{Z}$). The resulting equivalence class is the *pitch class*.

Scholars since at least 1704 have proposed geometric models that integrate absolute pitch height and pitch class. Newton, in his book *Opticks*, first represented color on a color wheel, and linked this to pitch, as in figure 6 [139]. Donkin projects the pith axis on a spiral in a polar coordinate system, as in figure 7 [46]. Others have proposed representations that lie on the surface of a cylinder or cone, with pitch height along the central axis of the body and pitch class represented by the angle around this axis, resulting in a helix-shaped structure [100]. This enriched embedding has been adapted by Chew as the "spiral array" model of tonality, and used for visualisation purposes and to define harmonic distances [33].

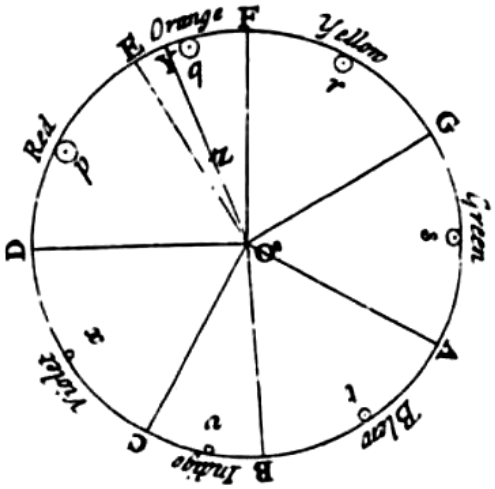

Figure 6.: Newton's circle representation of color and pitch. From [139].

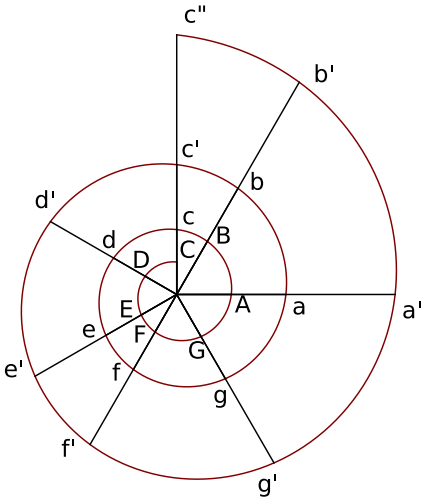

Figure 7.: Donkin's spiral representation of pitch. Adapted from [46].

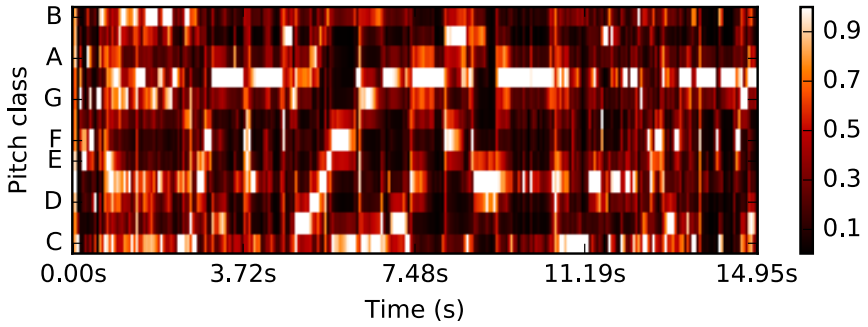

Figure 8.: A chroma time series.

Chroma Features

Chroma features are a representation of pitch class content that was first introduced by Fujimishima [54], as a feature for chord recognition. In its most basic form, chroma features or *pitch class profiles* (PCP), are a folded version of the constant-Q transform, in which the energy for all octave-equivalent pitches is summed together. Chroma features thus discard some of the absolute pitch height information, and only look at the pitch class dimension of the above spiral representation. Figure 8 shows a chroma time series computed this way.

Important advances in pitch description were made by considering the frequency-domain structure of complex pitches. The chroma features proposed by Gomez in 2006 do this by considering only prominent peaks in the spectrogram, summing together up to 8 harmonics per pitch and allowing for some deviation of the tuning frequency and the harmonic components, before folding the resulting harmonic pitch profile to a *harmonic pitch class profile* (HPCP) [55].

Wishing to reflect just pitch content and not the timbre information also present in the frequency spectrum, some pitch class representations take specific measures to achieve timbre invariance. Müller et al. discard timbre information by first computing the mel spectrum and setting the lowest 12-13 components of its DCT to zero before folding the amplitudes into pitch class bins [136]. This idea comes

from applications involving MFCC's, where it has been found that 12-13 coefficients are often enough to describe timbre in sufficient detail. Hence, keeping only the other coefficients might make for a good basis for timbre-invariant pitch class representation. HPCP features build in a similar invariance by "whitening" the spectrum prior to peak detection: a moving average of width m semitones is subtracted from the spectrum [55]. This approach is similar to Mullers, in the sense that both can be seen as a low-pass filtering operation on the spectral envelope.

Mid- and High-level Tonal Descriptors

The state-of-the-art in pitch and pitch class description involves a wealth of derivative features, most of which are built on the above representation. We review three families of mid- and high-level tonal descriptors.

Firstly, chroma features can be used to compute an estimate of the *tonal center* (e.g. tonic, in Western harmony), *mode* (e.g., major, minor), or *key* (tonic and mode) over a given time interval. This is typically done by estimating the correlation of the chroma features to a profile of pitch class occurrences. Commonly used profiles are the diatonic key profile and Krumhansl's empirically established templates [55,100]. Another, related harmonic descriptor is the *key strength* or *tonal/key clarity*, the confidence of the key estimate.

Harte and Sandler define the *tonal centroid* as the projection of the chroma vector in a complex geometric embedding. The 6-dimensional embedding can be seen as something in between the 12-dimensional chroma and the 2-dimensional circle of fifths. The tonal centroid feature can in turn be used to construct the Harmonic Change Detection Function (HCDF), a measure of harmonic change that is useful in (chord) segmentation of audio fragments [65]

In [56], Gomez and Herrera use a number of HPCP-based features to study the differences between Western and non-Western tonal music. Two relate to scale and tuning: the *tuning frequency* (deviation from 440 Hz) and equal-tempered deviation (average deviation from

an equal-tempered scale based on the tuning frequency), and are extracted from an HPCP with a resolution of 10 cents (0.1 semitones). They also compute the *diatonic strength*, the maximum correlation with a standard diatonic profile. Lastly, they include the *octave centroid*, the average pitch height computed before folding the pitch profile into an HPCP.

2.1.4 *Melody Extraction and Transcription*

Traditionally in music theory, *melody* is seen as the prominent, monophonic sequence of notes that characterizes a tune. Providing access to this sequence of notes, given a recording, has been one of the most elusive computational challenges in music information retrieval research. It is fair to say that, in the general case, complete reliable extraction and transcription of the main melody in a mix remains a challenging and unsolved problem, with state of the art accuracies of around 70% for the best systems– in just the extraction step [168]. The problem can be broken down into roughly three core issues. Firstly, separating components of a polyphonic mixture of complex sounds is very difficult. Mathematically, it is often an underdetermined problem, as the number of components is typically higher than the number of mixes (the latter usually being two in the case of stereo recordings). Humans solve this, to some extent, by employing a significant amount of top-down processing, i.e., using prior and contextual knowledge. Artificial systems can approximate these learned, contextual cues, but work on this is still catching up on the rapidly advancing technology coming out of the learning systems field. Secondly, estimating pitch from the separated stream is still a challenge in itself. To do this right, a system needs to decide when the melody is present and when it's not (voice detection), determine what note an octave is in, and pick out the main melody when notes overlap and sound simultaneously. Thirdly, identifying note boundaries remains a challenge for several instruments in which onsets (the beginnings of notes) and note transitions are blurred during performance (e.g., due to ornamentation) or production (e.g, reverb), including the singing voice. As the singing

voice often makes up the main melody, note segmentation on the main melody generally remains difficult.

We now review some pitch estimation systems based on three important strategies: the YIN algorithm (a time-domain approach) the Melodia algorithm (a frequency-domain approach), and an approach based on source-separation.

YIN

The YIN algorithm, proposed by de Cheveigne and Kawahara in 2001, is a pitch-estimation algorithm that operates in the time domain, i.e., no Fourier analysis is performed [40]. Instead, YIN works with a variant of the autocorrelation function (ACF) of the signal. The autocorrelation function scores the similarity of a signal with a time-delayed copy of itself. Formally:

$$\mathrm{ACF}(Y)(\tau) = \sum_t y(t)\, y(t-\tau) \tag{11}$$

The time delay τ is referred to as the *lag* and expressed in seconds. If a signal is exactly periodic with period T,

$$y(t) = y(t+T) \quad \forall T \tag{12}$$

the ACF will be maximal at $lag = T$:

$$\iff \mathrm{ACF}(Y)(T) = \sum_t y(t)\, y(t-T) = \sum_t y(t)^2. \tag{13}$$

but also at every multiple of T, including zero, as shown in figure 9. If the signal is noisy, but roughly periodic, the ACF will still reach a peak at every multiple of the period, so the method used to find T amongst these peaks needs to be considered carefully. Earlier systems based on the ACF were proposed by Hess and de Cheveigné [67]. Since, even with some measures in place, the autocorrelation method makes "too many errors for many applications", a cascade of error-reducing steps are added to the above procedure, to ensure that a sensible periodicity is found even if the signal isn't perfectly periodic, as is the case with

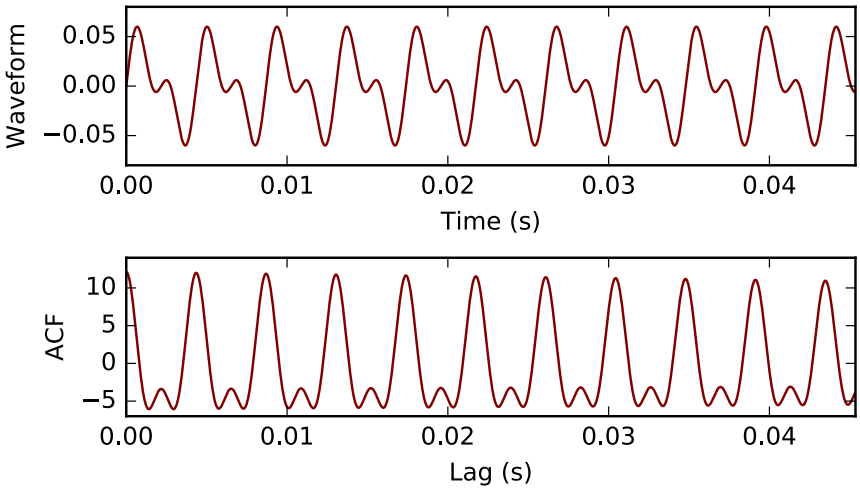

Figure 9.: A periodic signal and its autocorrelation function.

speech and musical pitch [40]. For example, instead of the ACF, the very similar autodifference function is found to increase performance:

$$\mathrm{ADF}(\tau) = \sum_t (y(t) - y(t-\tau))^2. \tag{14}$$

Finally, in another important modification to standard ACF-based approaches, over- and underestimates of T are avoided by not just looking for the global minimum of τ, but looking for the first value of $\mathrm{ADF}(\tau)$ that is substantially (e.g., 90%) lower than the average of ADF up to τ. This helps assess the significance of dips in the ADF near $\tau = 0$, where values are very low, as compared to dips for greater τ. Together, these measures constitute a robust procedure that has been shown to work for both speech and music. However, the signals on which YIN is typically used are largely monophonic, with only one, or one very dominant pitch present.

Melodia

Since the YIN algorithm, many have worked on main melody extraction from polyphonic signals: signals with several pitched sound sources present. Many of these systems start from a frequency representation of the signal, and assess, in different ways, the *salience* of all possible candidate pitches. One such system, which also performs well in comparative evaluations, is the Melodia system by Salamon.

Melodia pitch extraction is based on a high-resolution STFT, from which peaks are found. To get the best estimate of the exact location of these peaks, the *instantaneous frequency* is found by not just considering the magnitudes of the spectrum in each frame, but also interpolating between the phases of peaks in consecutive frames of the STFT [168]. Much like with HPCP (see 2.1.3), harmonic summation is then performed on the set of peaks (rounded to 10 cent bins) using a cosine weighting scheme. Extensive processing is also applied: peak candidates are grouped in time-varying melodic contours using a set of heuristics based on perceptual streaming cues. These candidate contours are then given a score based on their total salience and shape, and post-processed. The algorithm finally selects the set of contours that most likely constitutes the melody. In the latter step, the Melodia system also characterizes each contour as either voiced (sung by a human voice) or unvoiced, and decides in which frames no predominant melody is present at all.

The use of a contour representation has proven useful outside the core tasks of pitch estimation. In [169], Salamon and Rocha propose a number of mid- and high-level melody descriptors based on the contours extracted by Melodia, for a genre classification experiment. They include contour duration, mean pitch height, pitch height deviation and range, and presence and amount (width, frequency) of vibrato in each contour. Finally, each contour is also characterized as 1 of 15 contour types proposed by Adams, based on the order in which the highest, lowest, first and last pitch appear [2].

Data-driven and Source Separation-based Systems

A third group of melody extraction algorithms extract the melody by separating it from the rest of the mix. The simplest ones use a trained timbre model to describe each of two sources, one being the melody and another the accompaniment. These timbre models can be Gaussian mixture models (GMM's), in which each source is seen as a weighted sum of a finite set of multidimensional Gaussians, each describing a particular spectral shape, or hidden Markov models (HMM), a generalisation of GMM's. The models can be trained on source-separated ground truth data using expectation maximization [141]. Once the models have been used to separate the melody from the accompaniment, pitch estimation on the melody component is greatly simplified and a time- or frequency domain algorithm can be used. Advances were also made to apply newly developed machine-learning technology in the source separation step. In [180], Simpson et al. presented the results for a melody separation algorithm based on a deep convolutional neural network. The neural network is trained on spectrogram snippets of size 20×1025, yielding around a billion (10^9) parameters in total. The neural network approach improves on identifying main vocal melodies over a more traditional Non-negative Matrix Factorization-based approach. A purely data-driven model is presented by Poliner and Ellis. Skipping the separation step altogether, they use Support Vector Machines to classify STFT frames directly into melodic pitch categories [160]. The same authors have reviewed a number of other approaches in [161]. With the current trends in data-driven information retrieval methods trumping the performance of older, model-based approaches, more progress from this kind of methods can be expected in the near future.

2.1.5 *Psycho-acoustic Features*

Some of the features used in MIR relate to well-established psycho-acoustic qualities of sounds, e.g., loudness, sharpness and roughness. Each of these features quantises a perceptual attribute of sounds as

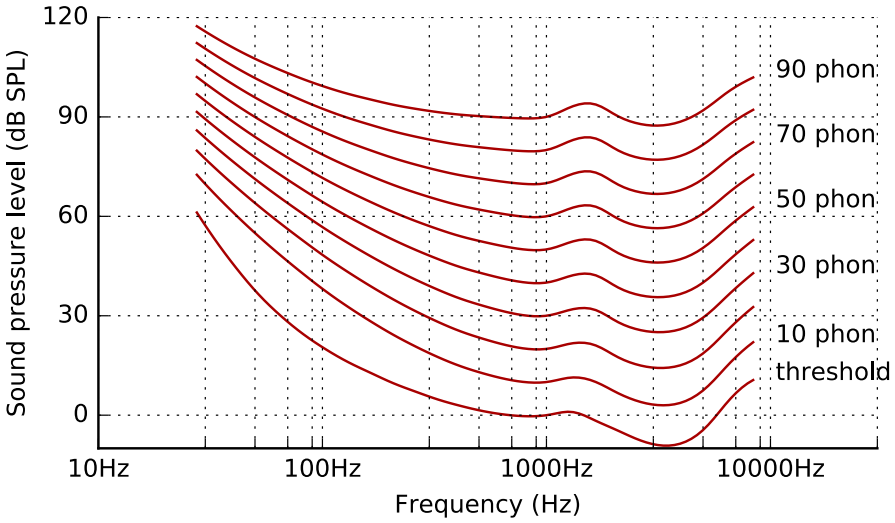

Figure 10.: Equal-loudness contours as specified by ISO standard 226.

rated by the participants of a listening experiment, or is based on (a model of) such measurements. The features take into account the non-linearities of the human auditory system.

The perceived *loudness* of a sound is determined by its intensity (as measured in dB) and its frequency content. At frequencies outside the 20–20,000 Hz range, sound is generally inaudible. But within the audible range, loudness varies with frequency as well. Equal-loudness contours specify this relation quantitatively. With these empirically established contours, shown in figure 10, a sound's loudness can be computed from its intensity at different frequencies. Two units of loudness exist: the *sone*, and *phon*. A single-frequency 1000 Hz sound at 40 dB has a loudness of 1 sone, and doubling a sound's perceived loudness doubles its value in sones. The phon is the basis of the ISO standard scale (shown in red in figure 10). A 60 dB SPL sound has a loudness of 60 phon. The phon scale is logarithmic: doubling a sound's perceived loudness adds 10 phon.

Loudness can also be computed for individual bands along the frequency spectrum. This yields an array of *specific loudness* values. A

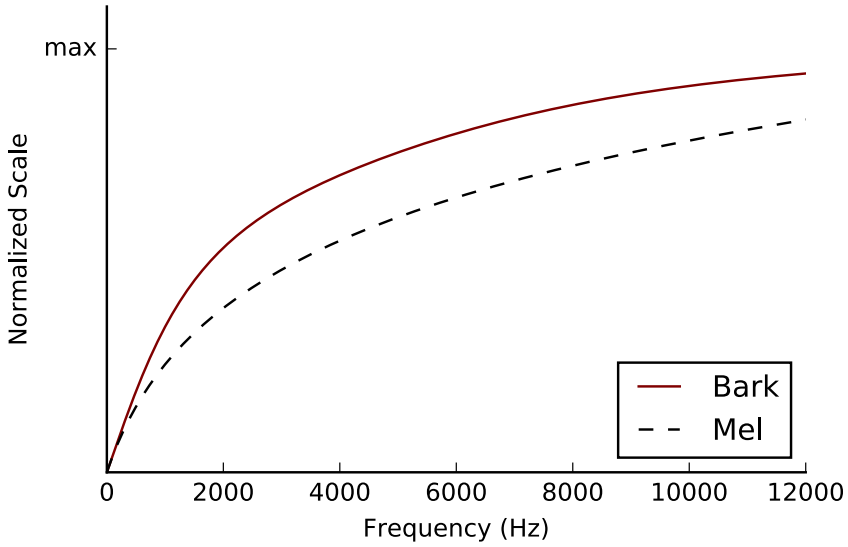

Figure 11.: The Bark and Mel frequency scales compared.

possible set of bands used for this purpose is the Bark scale. It is based on the mechanics of the inner ear. The arrangement of neurons along the inner ear's *basilar membrane* determines a *critical* bandwidth for every frequency. Within this band, masking occurs: the presence of one sound makes another more difficult to hear [111]. The Bark scale aims to take these elements of the frequency dimension's topology into account. Like the critical bandwidth and the somewhat simpler Mel scale, it is roughly linear at low frequencies (below 1000 Hz) and logarithmic at high frequencies (above 1000 Hz), as shown in figure 11.

Perceptual *sharpness* is a psycho-acoustic feature that is based on the Bark scale. While the above 'total' loudness integrates the specific loudness over all Bark bands, the sharpness feature measures the specific loudness distribution's centroid (i.e., center of mass). A sound for which the frequency content is more concentrated in the highest bands will have a high sharpness. Perceptual *roughness* is a result of

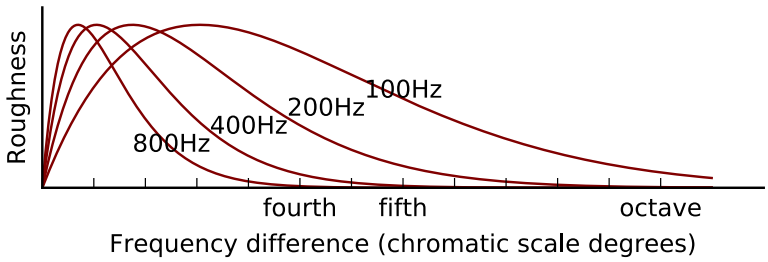

Figure 12.: Roughness as a function of frequency and frequency difference. Frequency difference is expressed in semitones.

the proximity of a sound's non-masked frequency components within the same critical bands. The roughness feature integrates the effect of these distances over the entire frequency spectrum. Somewhat simplified:

$$R(X) = \sum_{f_i} \sum_{f_j} w(f_i, \left|f_j - f_i\right|) X(f_i) X(f_j) \tag{15}$$

where w is a function of the first frequency f_i and its distance to the other frequency f_j over which is summed for every f_i. An example of different w for $f_i = 100, 200, 400, 600, 1000$ is shown in figure 12. Perceptual roughness is low for primarily harmonic, sinusoidal sounds and high for noisy and inharmonic sounds [159].

The above features not only correspond to empirically established attributes of sound, the attributes to which they correspond are also widely used in natural language description of sound. We argue that this makes them effectively high-level features. An analysis in which a trend for any of these descriptors is observed, can easily be translated back to domain language and natural language, making them an excellent instrument for computational research on musically motivated research questions.

2.1.6 *Learned Features*

As discussed in the beginning of this section, it can be useful to learn new representations entirely from data. This section reviews a number of techniques that can be used to do so, focusing on studies that don't just solve a particular task, but yield useful representations along the way. As with most trained systems, we can distinguish between supervised and unsupervised statistical learning.

Supervised learning generally requires a dependent variable that the learning system is trained to predict using a ground truth. New features can be constructed by taking the feature transformations that are learned in this process out of the trained system, to apply them somewhere else. For example, a multilayer neural network can be trained to predict labels for a set of labeled training data, so that, after it has been trained, one of the hidden layers can be used as more informative feature vector instead of the feature vector that was used as input, to address a different task. This is often referred to as transfer learning. In [193], a set of non-linear transformations of the STFT is learned by training a model to predict music listening statistics. The resulting representation is then successfully used in a number of different tasks and different datasets, showing that representations learned for one task can indeed be useful in a different context.

In unsupervised learning, the structure of an "unlabeled" training dataset itself is exploited to construct alternative representations. For example, a sparse auto-encoder is a neural network that learns a non-linear feature transformation. The new feature is taken out of a hidden layer (typically the only one) in this neural network, but instead of training the net to predict labels, it is trained to reconstruct its input while maintaining a sparsity constraint on the activations of the hidden layer. In [74], Humphrey et al. use non-linear semantic embedding (NLSE), a similar technique based on convolutional neural networks, to organize instrument samples in a low-dimensional space.

Dimensionality reduction and clustering techniques like PCA and K-means can be used as feature representations, too. In [35], Coates et al. showed that encoding a feature vector as an array of distances to

k cluster means can outperform other unsupervised feature learning techniques of similar complexity. This K-means approach has the advantage of having only one hyperparameter (k), and being efficient to train. Dieleman and Schrauwen applied a K-means representation to a tag prediction problem in [44].

As PCA and K-means are conceptually related, similar experiments have been done for PCA-based features. In [64], Hamel introduced principal mel spectrum components (PMSC). PMSC features are obtained using feature whitening and PCA on short arrays of mel spectrum frames.

Because of their probabilistic nature, statistical and neural network-based methods may appear to carry a suggestion of cognitive plausibility. Indeed: in a purely connectionist, statistical learning-centered perspective on cognition, learned features are a technology that may be, at the same time, optimal computational solutions to an engineering problem, and plausible cognitive models. However, not only is a purely connectionist view on music cognition generally disputed, current feature-learning methods are still far removed from realistic biological models of the brain, despite the quick successions of trends suggesting rapid progress.

2.2 APPLICATIONS OF AUDIO FEATURES

This section will provide a high-level overview of the music information retrieval field, focused on the problems, or tasks, researchers have addressed. We illustrate some of the most common practices in audio-based MIR research, to contextualize the origin of many of the features described in the previous section, and to give the necessary background for the critical discussion of these features in the next chapter. Most of the discussion, however, will be focused on those topics that are most relevant to this thesis.

In section 2.2.1, some of the most important work regarding music classification is reviewed, perhaps the core of 'classic' MIR, including the popular topics of genre and mood extraction from audio. In the next subsection, we review the most important methods in mu-

sic structure analysis, audio thumbnailing and chorus detection. This will be relevant to our work in Chapter 4. In section 2.2.3, the state-of-the-art in cover song detection and audio fingerprinting is reviewed. This will be relevant in Chapters 5 and 6.

2.2.1 *Audio Descriptors and Classification*

Audio classification tasks make up a large part of the most widely practiced research activities in music information retrieval, so they cannot be left out of a review of audio descriptor applications. But classification is also relevant because it can be seen as a form of high-level description, e.g., in terms of sociocultural information about the music. The resulting labels, then, are not properties of the music itself, but provide useful, user-level information for a variety of practical applications.

Genre classification

As one of the most widely researched topics in music information retrieval, genre classification deals with the automatic labeling of songs with genre tags. The appeal of this kind of information retrieval is easy enough to explain: originating in music sales and retail, genre tags provide a level of description that is useful in commercial contexts, and unlike many other descriptors used in MIR, genre and style labels are also widely used and understood by non-specialists [7].

The problem of genre classification allows for a very standard classification set-up: each document can be assigned one of a small set of class labels, and for each class a large set of examples can be found to train and evaluate classifiers on. Naturally, this involves some simplification, as genre description can be more or less detailed, and border cases are numerous.

To discuss all the audio features and classification algorithms that have been used in MIR would make for a very long and boring review. Most popular classification algorithms, like nearest neighbor classifiers, decision trees, support vector machines, neural networks

and random forests, have at some point or other been used to classify songs into genres [60].

One of the first and most influential studies to address music genre recognition (MGR) in depth is a series of experiments by Tzanetakis and Cook [188]. The study presents the first version of a now widely used dataset, *GTZAN*. A set of audio descriptors for MGR is proposed that includes tempo and rhythm features, timbre features (including MFCC), and some summary features computed from the pitch histogram. The classifiers that are studied are two classic density estimation models (a simple Gaussian model and Gaussian mixture models; GMM) and a non-parametric model (k nearest neighbours).

Since 2002, several improvements and variations were proposed that stick with the general approach of hand-crafting features and training a classical pattern matching classifier on the GZTAN ground truth, many of which were reviewed by Scaringella in 2006 [172] and Guaus in 2009 [60]. Notable additions to the above pipeline include features that build on improved models of the auditory systems, such as in the work by Panagakis [143], and the use of more powerful classification algorithms that have since emerged, such as support vector machines [115] and AdaBoost [11].

When, around 2010, feature learning techniques became widespread, MGR did not stay behind, and a variety of genre recognition systems were proposed that made use of technologies like learned sparse representations (e.g. [144]) and deep belief networks, a flavour of neural networks that are trained in a largely unsupervised manner [43].

Following these advances, classification accuracies reported in recent MGR studies have approached and exceeded the 90% mark on the GTZAN dataset [185]. It may be tempting to conclude that MRG is a solved problem, but as accuracies exceed even the GTZAN's theorized upper bounds due to inter-annotator disagreement, such claims taken with a grain of salt. This performance paradox has been explained by a combination of dataset issues (faults in GTZAN) and more fundamental issues around the usual approach to genre modeling, some of which will be discussed in section 3.3.3 [185].

The high performance numbers obtained in MGR may explain why feature learning researchers moved on to similar, but more difficult tasks, such as the more general 'tag prediction' task, with successes reported for convolutional neural network-based approaches and 'shallow' learning techniques such as k-means [44, 63].

Tag Prediction

In tag prediction experiments, a system is trained to predict manually assigned descriptive 'tags' for a dataset of songs. Contrary to MGR, tags can refer to any aspect of the music, including genre and style, but also instrumentation, language, topic of the lyrics, sentiment, geographic origin, era, mood, artist gender and form.

The rise of tag prediction and or 'social tag' prediction as a task can be traced back to the rise of the social web, where, on sites like *Last.fm*[1] and *MusicBrainz*[2], the enrichment of on line music data was crowd-sourced—by linking to social networks or through a Wiki-like platform.

Mood and Emotion Prediction

Another widely researched set of so-called top-level descriptors of music, are music mood and emotion. Music, in many parts of the world, is understood to fulfill a role as a 'tool' or medium for emotion regulation [71]. Application-oriented research efforts see emotion as an important practical attribute of music, that can be used in music search, recommendation, and in contexts like advertising and mood regulation apps.

Exactly how emotions are associated with specific pieces of music is a subject of debate. In music emotion literature, two mechanisms are typically distinguished. On the one hand, music can, to some extent, *express* emotions, through the intentions of the composer or performer. Emotional 'content' can then be perceived by the listener, though this perceived emotion isn't necessarily the same as what the artist intends

1 http://www.last.fm

2 http://www.musicbrainz.com

to express: the expressed emotions may not be perceived, or only selectively, while some of the emotional content perceived by the listener may not be intentionally communicated by the artist at all. On the other hand, there is the induced or felt emotion. This is the emotion that is induced in listening, and may be vastly different again from the emotion expressed by the artist or perceived to be expressed by the listener.

Much of the emotional value perceived in, and induced by music, is understood to originate externally to the music itself, in the listeners personal and cultural associations for example, or their social environment. This makes the task of automatic music emotion recognition (MER) from purely musical data difficult. A related task that circumvents the semantic subtleties of discussing emotion in music, is mood prediction. In this task, songs have been tagged with 'mood' labels, and a system is trained to reproduce these annotations. Whether mood prediction refers to a distinct task or merely a reformulation of perceived/induced emotion recognition, is beyond the scope of this discussion.

Many studies in MER work with an emotion space of just two dimensions: the valence-arousal plane, shown in figure 13 [206]. The idea is that all the emotions in this model can be situated along two principal axes. *Valence* relates to pleasure and distinguishes between positively and negatively experienced emotions. *Arousal* relates to energy or activation. Happiness, in this model, is a high-valence emotion characterized by a more or less average arousal. Anger is a low-valence, high-arousal emotion. Algorithms that use this space to model emotion simply need to predict both variables on a continuous scale, to describe a wide range of emotions, e.g. using regression models. Studies that have advanced this *dimensional* approach include work by Korhonen, Yang and Panda [96, 145, 207].

Other researchers have followed a *categorical* perspective on emotion. In this view, as well as in mood prediction, emotions and moods are treated much like tags: for each mood, a classifier or ensemble of classifiers is trained to predict its presence. The computational advantage of the valence-arousal models is that only two variables need to

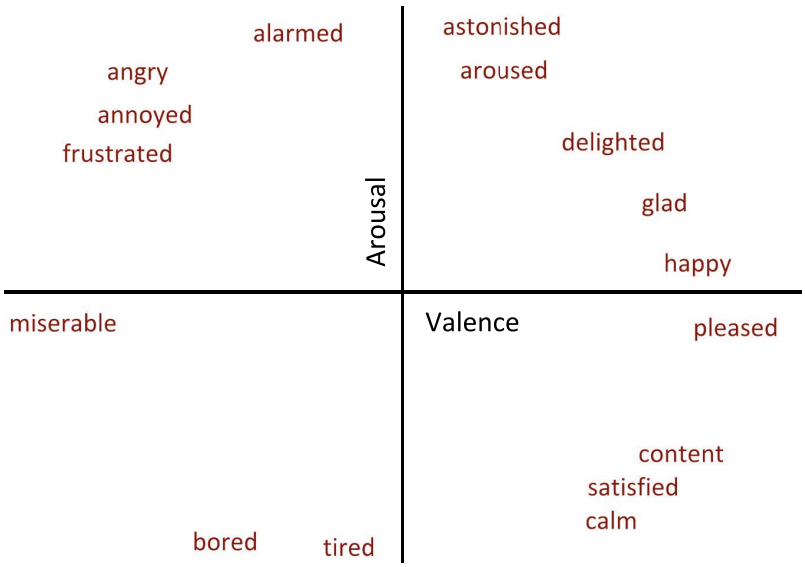

Figure 13.: The valence-arousal plane, a dimensional model of emotion (from [145]).

be modeled, compared to the very many (binary) variables typically involved in mood or tag prediction. The advantage of the tag prediction approach, however, is that the vocabulary doesn't have to be reduced to an agreed-upon space: mood tags might be include that do not seem to fit onto the plane at first sight (e.g., 'funny') or seem to collide (e.g., anger and fear) [91].

The first study to follow a categorical approach to mood and emotion recognition was done by Li and Ogihara and used 13 categories [109]. The audio mood classification task at MIREX uses a mood adjectives taxonomy based on 5 clusters. Others have used 4, 6, 8 and 18 clusters, to name just a few popular choices [206]. One music emotion taxonomy that allows for several 'resolutions' is the Geneva Emotional Music Scales (GEMS) model, a domain-specific model that was developed for music and allows for 9, 25 or 24 terms to be used. It was used in a study of induced emotion by Aljanaki in [4].

Summary

Throughout the many studies in MGR, MER and tag prediction, a number of recurring technologies and practices have come to fruition, often relying on a combination of audio features and classification schemes to reproduce high-level manual descriptions. Along the way, MGR and MER focus areas have carved out a practical and versatile approach to high-level music description—genre, tags, mood—that is powerful, but heavily reliant on machine learning.

2.2.2 *Structure Analysis*

In the MIR field of audio structure analysis, tools are developed to extract information from audio files on the level of structure or form. Commonly with the intent of using this information for further processing; in a few cases, as an end goal. For example, some applications of MIR benefit from prior segmentation of a recording, e.g. audio similarity computation [165]. Structure analysis as an end goal can be seen in a service like the Music Listening Station by Goto et al. [58].

Specific structural information retrieval tasks include structural segmentation, phrase segmentation, summarization, thumbnail extraction, chorus detection and full structure analysis. Structural segmentation refers to finding the boundaries of structural sections. In thumbnail extraction and music summarization, a stretch of audio is reduced to a one or more short subsections that are maximally representative of the recording [10]. Chorus detection refers to a similar task for popular music, in which the chorus of a song is located, to be used for indexing or as a representation in a browsing interface [57]. Structure analysis typically refers to structural segmentation, followed by a labeling of each segment with their structural function (e.g. verse or chorus in popular music, head in jazz, stanzas in folk music, exposition and bridge in classical forms, etc.) [38].

Two good overviews of structural analysis in the literature are provided by Dannenberg and Goto in [38] and by Paulus, Klapuri and Müller in [150]. The latter distinguishes between novelty-based, homogeneity-based and repetition-based approaches, echoing a distinction made first by Peeters in 2007 [155]. Peeters identified two general strategies: the state approach and the sequence approach. Most of the research follows one of these strategies; a few attempts have been made to combine both approaches. The most important contributions will now be explained, following Peeters' distinction.

State-based Structure Analysis

In the *state* approach on structure analysis, a song is interpreted as a succession of observable states, which can be mapped to structurally meaningful sections or 'parts'. A state spans a contiguous set of times during which some acoustical properties of a song are more or less constant. This is said to hold for popular music, in which the 'musical background' often remains the same throughout a structural section. The state approach is applied mostly in combination with timbre features, such as MFCC, since they tend to correlate with instrumentation [150].

State representations can be obtained in various ways. The *novelty approach*, for example, detects transitions by looking for peaks in a novelty function. A naive novelty function can be constructed by correlating a feature time series with a length N novelty kernel such as

$$z(n) = sign(n) \cdot \Phi(n), \quad n = -N/2 \ldots N/2 \tag{16}$$

where Φ is some symmetric Gaussian. Other approaches apply HMM or similar methods to group frames of features into states, often using two (or more) techniques sequentially. Clustering the obtained states finally allows for the mapping of the observed time spans to more or less meaningful musical parts.

Responding to the field-wide growing appeal of data-driven methods, Ullrich et al. recently proposed a relatively simple method using convolutional neural networks (CNN) [190]. A CNN is trained to predict the presence of a boundary given a region of frames of a basis feature, with good results. The network does not keep any history, so the cues it uses can be assumed to relate to novelty rather than repetition, making this method effectively a state-based one.

Sequence-based Structure Analysis

The sequence approach relies on repetitions of sequences of features to infer structure. The frames in a sequence do not need to show any similarity amongst themselves, as long as the sequence as a whole can be matched to a repetition of the sequence somewhere else in the song. Most repetition finding approaches work on the *self-similarity matrix* (SSM) of the song.

The self-similarity matrix is an essential tool of many structure analysis algorithms. For music feature representations that are computed over short frames, the *self-similarity matrix* (SSM) is a matrix representation of the similarity of each frame to every other frame:

$$\text{SSM}(i, j) = s(X(i), X(j)) \tag{17}$$

where s is a function measuring similarity between two frames of audio. SSM matrices reveal state structure as homogeneous blocks, while

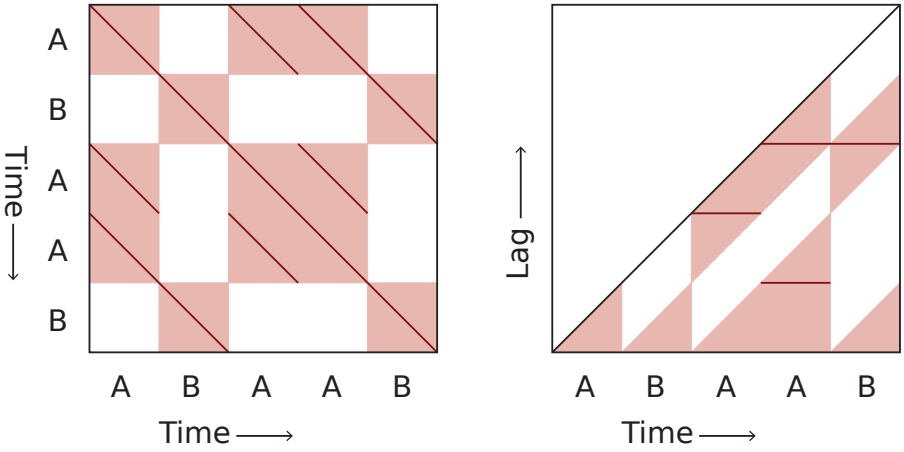

Figure 14.: An idealized SSM and corresponding time-lag matrix. Darker regions denote higher similarity. The state structure can be seen as blocks while the sequence patterns are visible as 'stripes'. Adapted from [150].

repeated sequences emerge as diagonal 'stripes' as shown in figure 14 (left). Sequence-based approaches using SSM focus on these stripes, and use chroma as its basis feature $X(t)$.

Variants of this approach include the use of a *self-distance matrix* (SDM), which contains the distance between two frames d rather than the similarity, or the *recurrence plot* (RP), in which a short history of frames is used to assess similarity. Common distance functions are simple Euclidean and cosine distances.

The SSM's *time* × *time* representation of self-similarity can be converted to a triangular *time* × *lag* matrix (shown in figure 14 on the right) through appropriate 'folding', i.e. $(i, j) \mapsto (i, i - j)$, in which the lag $i - j$ is the time difference between frame i and j. *Time* × *lag* matrices conveniently show repetitions as vertical stripes. Some methods make use of a beat-synchronous SSM to account for tempo variations within the song, or the transposition invariant SSM introduced by Clausen and Müller [34].

Finding repeated sequences in the SSM is not as straightforward as it may seem and greatly benefits from the post-processing of the SSM after it has been computed. A moving average filter can be used to smoothen the matrix along the columns or diagonals as in [9, 57], as well as erosion and dilation (two grey-scale image processing operations, often combined to remove short interruptions in a uniform sequence) [50,114]. A high-pass filter can in turn be used to emphasize details in the opposite (lag) direction. The result is then typically converted to a binary SSM by comparing to a constant or relative threshold.

Finally, a set of repetitions is extracted from the SSM, each identified by a start, end and lag time. Two strategies can be observed. Goto's RefraiD algorithm [57] extracts repetitions by first looking for those lag times corresponding to the lowest distances. Along these columns or rows, it then stores all appropriate length time intervals in which the binary distance value is zero. Unless the feature frames are beat-synchronous, this method doesn't allow any deviations in timing or tempo. Another method proposed by Chai [32] uses dynamic time warping (DTW, see section 2.2.3) for the alignment of sequences to account for local tempo variations. Computationally, this is not very efficient since many different sub-matrices need to be matched (one for each pair of candidate sequences). Variations based on dynamic programming were used by Dannenberg and Hu [39], Paulus and Klapuri [147, 149] and recently by Müller, Grosche and Jiang [137].

Combining State and Sequence Representations

A number of recent methods have combined steps from these state and sequence approaches to advance the state of the art in structural segmentation accuracy. In [85], a simple combination method is proposed for the fusion of two independently obtained sets of candidate boundaries.

A more sophisticated method was proposed by Serrà in [175]. In this method, a newly proposed variant of the lag matrix is filtered along the time axis with a step function kernel, with the aim of de-

tecting the start and end points of individual repeated segments. The resulting 'structure features' matrix is then summed over the lag axis to obtain a novelty curve quite like the one typically used in state-based segmentation methods. The method is reported to work for both timbre and pitch features, and performed better than its competitors in the MIREX 2012 audio structure analysis track. Peeters et al. developed this idea further by combining it with a prior analysis following Goto's approach described above [156].

Another method proposed by McFee in 2014 integrates state- and sequence-based methods using a graph representation of the audio, rather than an SSM [125]. Each frame of the base feature X is represented as a node in a large graph, and edges between nodes are weighted by the similarity between the frames. A technique called spectral clustering is then applied to obtain a hierarchy of section boundaries, from which the best set of boundaries can be obtained in a supervised way, by letting a user or 'oracle' select the most appropriate level of segmentation.

Thumbnailing and Chorus Detection

As implied in the introduction to this section, summarization and audio thumbnailing are practically the same task. The term thumbnailing was introduced by Tzanetakis in [189]. Chorus detection is de facto a form of thumbnailing, specific to popular music, in which one wishes to locate the chorus of a song.

Definitions of chorus often make sure to include that a chorus is *prominent* and/or *catchy*, though this is rarely explained or formalized. Both thumbnail and chorus are essentially reduced to the *most often-repeated segment*. Just like in much of structure analysis, most research is therefore devoted to finding these repeated segments, combined with minor heuristics limiting the candidates. More advanced approaches include the system by Goto [57] and Eronen [50].

Any of the repetition-detection methods discussed above can in principle be used, and many of them come from papers on summarization and chorus detection. We conclude this section with an overview

of the heuristics that are used to select the most representative repetition.

After the obtained repetitions are 'cleaned' (using some heuristics for boundary refinement and dealing with overlap), they may be clustered to obtain meaningful groups, each corresponding to a part of the song, like in full structure analysis. Transitivity may be exploited here: if b is repetition of a and c is a repetition of b, then c should be a repetition of a. The grouping task, especially important in full structure analysis, is not trivial either. Cost functions and fitness measures have been proposed to rate the amount to which a proposed structure explains the observed patterns [137, 149, 155].

Scoring functions for the assessment of thumbnail and chorus candidates have been proposed as well [50, 57], introducing a variety of heuristics. The RefraiD system by Goto [57] makes use of a scoring function that favors segments c occurring at the end of a long repeated chunk abc and segments cc that consistently feature an internal repetition. Eronen [50] favors segments that occur near $1/4$ of the song and reoccur near $3/4$ as well as segments with higher energy. In most cases, heuristics are only used to limit the candidates from which the most frequent segment is picked. For example, by considering only the first half of the song or discarding all segments shorter than 4 bars.

Regarding chorus detection, it is clear that existing strategies in chorus detection only attempt to locate refrains in a pragmatic way, and do not aim to model what choruses are and what makes them distinct. This problem is addressed as part of this thesis, and detailed in chapter 4.

2.2.3 *Audio Fingerprinting and Cover Song Detection*

Audio fingerprinting and cover song detection systems both deal with the automatic identification of music recordings.

Robust, large-scale audio fingerprinting was one of the first problem in music information retrieval to be convincingly solved, and developed into a successful industry product. Effective audio fingerprinting algorithms like the ones developed by Haitsma and Kalker at

Philips [62] and Wang and Smith at Shazam [199] can reliably identify a single exact music fragment in a collection of millions of songs. This is useful as a service: the Shazam algorithm stands as a very popular app, and was even available before smartphone apps, as a phone service. But the technology can also used for content identification of on line radio, and on social networking sites and streaming services like Youtube[3] and Soundcloud[4]. Last but not least, fingerprinting can also be used to manage large collections and archives, e.g. for duplicate detection.

In cover song detection, or (cover) version identification, a system is charged with the task of matching a recording to other known versions of the same musical work, generally interpreted by other artists. This can be useful in a similar set of applications, most notably content identification and duplicate detection, but also plagiarism detection and music recommendation [174].

Audio Fingerprinting

Audio fingerprinting, at its core, involves the reduction of a large audio object to a compact, representative digest. Given an unlabeled fragment, fingerprinting systems extract this fingerprint and match it to a large reference database. State-of-the-art algorithms for audio fingerprinting produce fingerprints with a high degree of robustness to noise, compression and interference of multiple signals, and perform matching of fingerprints very efficiently [31, 59].

The first widely successful fingerprinting technique was proposed by Wang and Smith, the so-called *landmark-based* approach [199]. Like most fingerprinting systems, Wang's system includes an extraction and a matching component. In the extraction component, a piece of audio is first converted to a spectrogram representation using the STFT, and the most prominent peaks in the spectrogram are detected. Peaks are then paired based on proximity. Pairs of peaks are called landmarks, and can be fully described by 4 parameters: a time stamp,

3 `http://www.youtube.com`
4 `http://www.soundcloud.com`

the frequencies of both peaks, and the time interval between them. In a last step, the two peaks frequencies and the time interval are combined into a hash code for efficient look-up.

The reference database is constructed by storing all hashes for a collection of songs into an index, where each hash points to the landmark start time and a song ID.

In the matching stage, when a query is passed to the system, its landmarks and corresponding hashes are computed as described above. Any matching landmarks from other songs are then retrieved from the reference database, with their corresponding start time and song ID. Note that this can be done in constant time. In the last step, the system determines for which landmarks the start times are consistent between query and candidate, and the song with most consistently matching landmarks is returned as the result.

Haitsma and Kalker's approach, developed around the same time, also relies on the indexing of structures in the spectrogram, but doesn't involve the detection of peaks at a high frequency resolution [62]. Instead, the spectrogram's energy is computed in 33 non-overlapping bands, and the resulting time series is differentiated across both time and frequency. The resulting 'delta spectrogram' is then binarized by considering only the *sign* of its values. In the extraction step, strings of 32 bits are extracted from this representation, and stored as sub-fingerprints, much like the landmarks in Wang's approach. The matching step follows a similar logic as well.

Alignment-based Cover Detection

Despite the conceptual similarity between audio fingerprinting and cover song detection, state-of-the-art audio fingerprinting and cover detection algorithms share very little of their methodology. This is largely due to the many invariances that need to be built into any cover detection technique: any two performances of a song may vary in just about every aspect of their musical material, and still be regarded cover songs [177]. A good identification system should there-

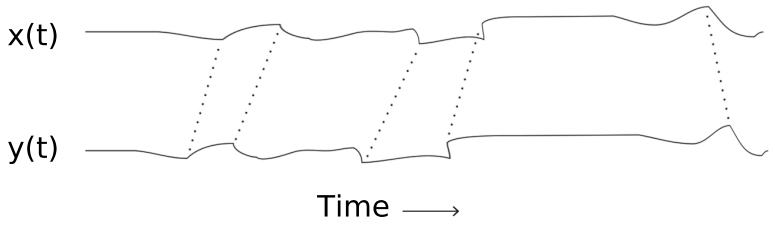

Figure 15.: Diagram showing the alignment of two time series performed in dynamic time warping. Adapted from [135].

fore be invariant to changes in tempo, key, structure, lyrics and instrumentation as well as, to some extent, melody, harmony and rhythm.

Cover detection systems have addressed this challenge in several ways. For example, tempo invariance, by representing songs as beat-aligned time series before matching [49], or key invariance, by performing a search using multiple, transposed, queries [102]. In [176], Serrà et al. propose a method based on the alignment of pairs of chroma time series that is not the first in its kind, but successful due to the incorporation of several novel invariance measures. Most notably, key invariance is achieved by first comparing the pitch histogram for each pair of songs, and transposing them to a common key. Tempo invariance is achieved using dynamic programming local alignment (DPLA) in [176], a form of locally constrained dynamic time warping.

Dynamic time warping (DTW) is an algorithm for the pairwise alignment of time series. In DTW, two time series are 'warped' in a non-linear fashion, to match each other in a maximum number of similar positions, as shown in figure 15. A score is assigned to each of several possible configurations, based on the similarity of matching frames and the number of skipped frames in each time series [135].

In [178], $Q_{\max}$ is introduced instead, a cross-recurrence measure defined on the cross-recurrence plot (CRP). The latter stems from non-linear systems analysis, and can be seen as a variant of the similarity matrix (SM) between two time series, where, like in the SSM (Section 2.2.2),

$$\mathrm{SM}(i, j) = s(X(i), Y(j)) \tag{18}$$

with s a similarity measure. The cross-recurrence plot differs from the standard similarity matrix by incorporating some 'history' of length m, the *embedding dimension*. This history is encoded in so-called delay coordinates:

$$\begin{aligned} \vec{X}_m(i) &= X(i-m:i) \\ \vec{Y}_m(j) &= Y(j-m:j) \end{aligned} \tag{19}$$

where : denotes a range. The CRP is given by:

$$\mathrm{CRP}(i,j) = s(\vec{X}_m(i), \vec{Y}_m(j)). \tag{20}$$

The cross-recurrence measure $Q_{\max}$ essentially measures how long the longest alignable segments are. Algorithms based on $Q_{\max}$ have performed best on the MIREX audio cover songs identification task since 2007 [178].

Scalable Cover Detection and Soft Audio Fingerprinting

While similar in concept, it is now clear that audio fingerprinting systems and cover song detection systems, as described above, are vastly different in their approach. This section will look at the commonalities to both strands of research, and lay out the prior art that combines ideas from both fields to address a common underlying problem. This work will be expanded on in chapters 5 and 6.

As stated at the beginning of this section, the common underlying problem between audio fingerprinting and cover song detection is the automatic content-based identification of music documents. Assuming a trade-off between efficiency and accuracy, we could say audio fingerprinting is an efficient solution to this problem, but not a very robust one: solutions are robust to several kinds of distortions to a query, but not to the wide variety of deliberate changes that cover detection systems take into account. They are unable to identify covers, live renditions, hummed versions, or other variations of a piece.

Conversely, the cover song detection systems reviewed above handle these modifications, but do so in a much less efficient manner. All of the cover detection systems reviewed in the previous section rely on some kind of alignment to assess the similarity for every pair

of songs. Since each query is linear in the size of the dataset N (N alignments are needed), and each alignment polynomial in the song lengths m and n, alignment-based algorithms are not a good solution for large-scale cover song retrieval [75].

Several efforts were made to adapt the concept of fingerprinting to such use cases, which require invariance to intentional, performance-related changes to the song. Relevant work includes a growing number of studies on 'scalable' cover song detection, pitch- and tempo-invariant fingerprinting, including sample identification, and some of the work done on 'query by humming' (i.e. identifying a song from a hummed or sung melody, generally performed by an amateur singer using a dedicated retrieval system).

In this thesis we refer to all of these tasks together as *soft audio fingerprinting* systems. The defining distinction between soft audio fingerprinting and other kinds of document retrieval, is the fixed size of the representations, which enables the use of an index to store them—guaranteeing the constant-time look-up.

Some of these soft audio fingerprinting systems follow Wang's landmark-based strategy, but build in some invariance. Audio fingerprinting systems targeting invariance to pitch-shifting and/or time-stretching include [51] by Fenet et al. and *Panaka*, by Six et al. [181]. Van Balen et al. [192] and Dittmar et al. [45] present automatic approaches for the identification of samples used in electronic music production. In each of these studies, the basis feature from which the peaks are combined into landmarks, is the constant-Q spectrogram, rather than the spectrogram.

Another landmark-based retrieval system is the large-scale cover song identification system proposed by Bertin-Mahieux et al. in [12]. Here, landmarks are extracted from pitch class profiles or chroma features (Section 2.1.3). As in fingerprinting, matching landmarks (here: 'jumpcodes') are retrieved with their song IDs. The study reports a mean average precision of about 0.03% and a recall of 9.6% on the top 1 percent of retrieved candidates in a large dataset: promising, but nowhere near the performance of alignment-based algorithms in their respective use case.

A more novel audio indexing feature, the *intervalgram*, is proposed by Walters [198]. It is essentially a two-dimensional histogram of local pitch intervals at various time scales, designed for hashing using wavelet decomposition. Another novel approach by Bertin-Mahieux uses a 2D Fourier Transform of beat-aligned chroma features to obtain a compact representation that is invariant to pitch shifting and time stretching [13]. This method was adapted by Humphrey et al. to include a feature learning component for more robustness to common variations [75]. The latter currently performs best in terms of large-scale cover song retrieval precision, though, with a mean average precision of 13.4%, still not close to the alignment-based state of the art.

The progress on some related tasks, such as query by humming, has been better. There is little information about the exact workings of commercial services such as Soundhound's MIDOMI[5], but they work well enough for commercial use. However, they are generally understood to rely on matching (alignment or otherwise) of simplified contours of melodies sung and labeled by volunteers, rather than matching hummed melodies with a song's original audio, which remains an unsolved problem.

2.3 SUMMARY

We have reviewed, in section 2.1, a selection of topics and state-of-the-art methods for audio description. We have focused on timbre description, harmony and melody description, psycho-acoustic features and learned audio descriptors. Several of these reviewed audio features and applications will be applied and improved upon in part ii of this thesis.

In section 2.2, we have reviewed a selection of the music information retrieval applications for which these features were developed. Here, we focused on classification and labeling tasks (genre, tags, mood and emotion), structure analysis, and music content identification (audio fingerprinting and cover song detection). In the last category, we have

5 http://www.midomi.com/

defined *soft audio fingerprinting* as the umbrella task of scalable music content identification, including efficient cover song detection, pitch- and tempo-invariant fingerprinting, sample identification, and query-by-humming. The task of soft audio fingerprinting is both an important open issue in MIR, and closely related to the project goal of modeling document similarity in musical heritage collections. Therefore, it will also be given more attention in the following chapters.