CHAPTER 3

INSTANTANEOUS FEATURES

Over the last few decades, a set of different widely used features has established itself for audio content analysis. Many of these features will be presented in this chapter. However, the choice of the specific features used in an algorithm will in the end always be driven by the task at hand; this can make the modification of well-known features as well as the design of new features advantageous or even mandatory. Therefore, the number of possible features used in audio content analysis is probably limitless and only a more or less representative set can be presented in the following. The various pre- and post-processing options closely related to feature extraction will be covered by this chapter as well.

The term *instantaneous feature*, *short-term feature*, or *descriptor* is generally used for measures that generate one value per (short) block of audio samples. An instantaneous feature is not necessarily musically, musicologically, or perceptually meaningful all by itself, and it is frequently referred to as a *low-level feature*. A low-level feature can serve as a building block for the construction of higher level features describing more semantically meaningful properties of the (music) signal (such as tempo, key, melodic properties, etc.).

A good example for an instantaneous feature is the magnitude or power level of the audio signal extracted on a block-per-block basis. It represents a widely known reduced representation of the audio signal. Although simple to extract, it may show characteristics usable for the extraction of higher level information. As Fig. 3.1 illustrates with three envelope excerpts of length 15 s, the signal categories speech, chamber music, and pop music show some variation that may be easily identified by an experienced user. This is, of course, not necessarily true for every possible excerpt, but it tells us that some kind of

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

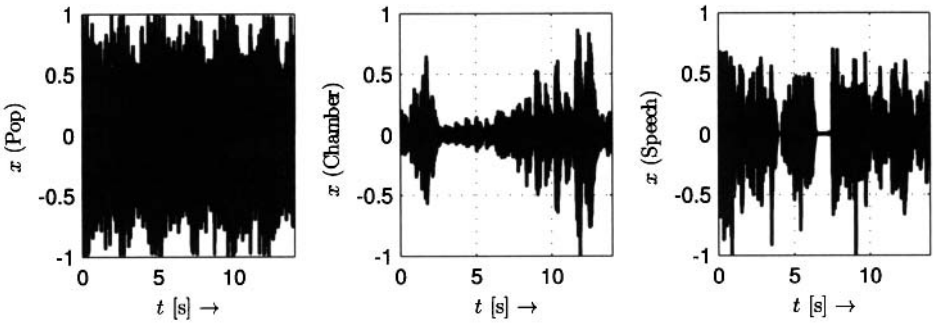

Figure 3.1 Envelope of excerpts from a typical speech recording (left), a string quartet recording (mid), and a pop recording (right) with a length of 15 s

envelope characteristic may be useful in automatic detection of a signal type or musical genre classification.

Instantaneous features can be categorized using different taxonomies. Probably the most obvious categorization is in the computational domain, which in the case of audio signals is usually either the *time domain* or the *frequency domain*. In several applications, features are calculated from other features, so the *feature domain* may be added as well. In the MPEG-7 standard, the following feature categories have been used: *basic*, *basic spectral*, *signal parameters*, *temporal timbral*, *spectral timbral*, and *spectral basis representations* [33]. Peeters categorizes features into the classes *temporal shape* (e.g., waveform-based envelope), *temporal* (e.g., zero crossing rate, ACF coefficients), *energy* (e.g. global or tonal energy), *spectral shape* (STFT-based features), *harmonic* (e.g., harmonic/noise ratio, tonalness), and *perceptual* (e.g., modeling human hearing) features [34]. Eisenberg uses the three categories *time domain*, *spectral domain*, and *harmonic* features [35].

As can be seen from these examples it is difficult to find a simple and consistent yet practically useful feature categorization. One feature may, for example, fit into more than one category because the categories may be overlapping. However, in the end this discussion is in itself only of limited use, so we will just boldly use the following categories for our instantaneous features in the following:

- *statistical properties*: features that are commonly used in statistical signal description such as standard deviation etc. (see Sect. 3.2),
- *spectral shape*: features describing the shape of the (magnitude spectrum of the) STFT (see Sect. 3.3),
- *technical/signal properties*: features that describe specific technical properties of the signal and cannot be categorized in other domains (see Sect. 3.4), and
- *intensity properties*: features closely related to the amplitude or intensity of the audio signal such as volume and loudness (see Chap. 4).

3.1 Audio Pre-Processing

The raw audio data is frequently pre-processed before computing instantaneous features from the data. The motivation for this pre-processing step is to reduce the amount of audio data to be analyzed by omitting unnecessary information or to minimize the impact of unwanted information on the extracted features. The standard approaches differ mainly in terms of whether the designed algorithm works in

- *real time*, meaning that only the current and past samples or blocks of samples are known, or
- *offline*, meaning that all upcoming samples of an audio file are known as well.

3.1.1 Down-Mixing

In most of the analysis problems the information of interest can be represented by one single audio channel as well as by multiple input channels. For example, the tempo or information on the musical style should be extractable from a mono-recording as well as from stereo or multi-channel recordings.

Down-mixing is usually done by simply computing the arithmetic mean as defined by Eq. (3.10) over all input channel signals $x_c(i)$ per sample i. The number of input channels is C.

$$x(i) = \frac{1}{C} \sum_{c=0}^{C-1} x_c(i). \tag{3.1}$$

It is also possible to apply different weights to different channels; surround channels may, for example, have a lower weight than front channels. An alternative to down-mixing could be to use only one pre-selected audio channel, however, this may result in loss of information if the channels have been mixed to produce a wide spatial image. Some audio applications also apply a phase shift of 90° to one channel before down-mixing a stereo signal to mono. The reason is to avoid a level boost of components present in all audio channels (mono-components).

3.1.2 DC Removal

A DC offset — shown by a signal's arithmetic mean significantly different from zero — usually does not provide any useful information and may have unwanted impact on the feature results.

3.1.2.1 Offline

If all audio data $x_{\mathrm{DC}}(i)$ of length $\mathcal{I}$ is available, it is possible to simply compute the arithmetic mean and subtract it from every sample:

$$x(i) = x_{\mathrm{DC}}(i) - \frac{1}{\mathcal{I}} \sum_{i=0}^{\mathcal{I}-1} x_{\mathrm{DC}}(i). \tag{3.2}$$

3.1.2.2 Real Time

A self-evident method to remove the DC part from your signal is to apply a high-pass filter to it. Basically any high-pass filter can be used. The simplest method to do so in real time

is to apply a differentiator:

$$x(i) = x_{\mathrm{DC}}(i) - x_{\mathrm{DC}}(i-1). \tag{3.3}$$

A differentiator, however, has significant impact on higher frequency components as well. This effect can be lessened by low-pass filtering the difference; the resulting DC filter would be

$$x(i) = (1-\alpha) \cdot \big(x_{\mathrm{DC}}(i) - x_{\mathrm{DC}}(i-1)\big) + \alpha \cdot x(i-1). \tag{3.4}$$

Another possibility is to use a long MA filter (see Sect. 2.2.1.1) of a length $\mathcal{O}$ which provides a sufficiently reliable estimate of the arithmetic mean to be subtracted from the input signal:

$$x(i) = x_{\mathrm{DC}}(i) - \frac{1}{\mathcal{O}} \sum_{j=i-\mathcal{O}/2}^{i+\mathcal{O}/2-1} x_{\mathrm{DC}}(j). \tag{3.5}$$

3.1.3 Normalization

In order to extract features independently of the amplitude scaling of the input signal, the signal can be *normalized* to have a pre-defined (maximum) amplitude or power.

3.1.3.1 Offline

A simple and frequently used method to normalize an audio signal is to detect the overall maximum of its absolute sample values and scale the signal so that this maximum's absolute value is mapped to 1:

$$x(i) = \frac{x_s(i)}{\max\limits_{\forall i} \big(|x_s(i)|\big)}. \tag{3.6}$$

This approach results in a normalized magnitude but does not warrant equal loudness of different input files. Furthermore, the *normalization* may be influenced by signal distortions such as the clicks and crackles of a vinyl recording. In this case, the normalization to the maximum click amplitude will result in an "incorrect" scaling of the music data.

The alternative is to use some other reference than the maximum for the derivation of the scaling factor such as an RMS (see Sect. 4.3.1) or a loudness measurement with large integration time (see Sect. 4.5). In this case, additional processing may be necessary in order to avoid potential clipping of the scaled signal.

3.1.3.2 Real Time

Normalizing a signal in a real-time context is difficult. It can be done with algorithms for automatic gain control or compressors and limiters monitoring the instantaneous input signal characteristics (e.g., the RMS or the peak level) and aiming to adjust a time-variant gain value according to it.

3.1.4 Down-Sampling

Down-sampling requires the application of a *sample rate conversion* algorithm which converts the input sample rate f_{S} to a lower sample rate f_{d}. Down-sampling thus reduces both the amount of audio samples and the bandwidth of the resulting audio signal.

The easiest way to down-sample is to reduce the sample rate by an integer factor l. The output sample rate is

$$f_\mathrm{d} = \frac{f_\mathrm{S}}{l}. \tag{3.7}$$

If we just pick every lth sample, then the down-sampled signal x_d is

$$x_\mathrm{d}(i) = x(l \cdot i). \tag{3.8}$$

Taking into account the sampling theorem as given in Eq. 2.9 and the time scaling property of the spectrum as given in Eq. (2.54) it becomes clear the aliasing will occur if the input signal $x(i)$ contains frequency components higher than $f_\mathrm{d}/2$. In order to ensure artifact-free down-sampling a low-pass filter has to be applied to the input signal to remove any frequency components higher than $f_\mathrm{S}/2l$. Sample rate conversion with non-integer factors is based on the same principles. The factor can be written as the ratio of two integer factors s/l so that the signal can first be up-sampled by factor s and the down-sampled by factor l. In between the two resampling steps it is required to apply a low-pass filter which ensures both the reconstruction of the up-sampled signal and the suppression of aliasing artifacts in the down-sampled signal. It is possible to use various combinations of interpolation algorithms and low-pass filters for down-sampling a signal. One example for such a *band-limited* interpolation has been presented by Smith and Gosset and is usually referred to as *sinc* interpolation [36].

3.1.5 Other Pre-Processing Options

As with the selection of appropriate features itself, the pre-processing options will always be adjusted to the application in mind. Every pre-processing which improves the algorithm's accuracy, its stability, or minimizes its computational workload is beneficial. In addition to the presented pre-processing options it is, for example, also common to attenuate the level of any unwanted frequency region by applying a filter to the signal.

3.2 Statistical Properties

Various methods describing the properties of a (time-invariant) properties of a signal are well-established and frequently used. These measures can be applied to both, the time-domain signal block as well as the spectrum. While the definitions below use $x(i)$ as input signal, it could be substituted by $X(k, n)$, by a series of feature values $v(n)$ or by any other signal of interest.

Theoretically, the statistical properties presented below require a signal of infinite length, however, in practical applications they can be assumed to be sufficiently accurate if the block is long enough. The block length will be denoted as

$$\mathcal{K} = i_\mathrm{e}(n) - i_\mathrm{s}(n) + 1. \tag{3.9}$$

In order to simplify definitions, there will be no differentiation between the theoretically correct property or measure (for the signal with infinite length) and its estimate (from a finite length signal block).

3.2.1 Arithmetic Mean

The *arithmetic mean* is the average of the input signal (block). It is computed by

$$\mu_x(n) = \frac{1}{\mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} x(i). \tag{3.10}$$

The result of the arithmetic mean is a value between the minimum and maximum input signal value. The unit corresponds to the unit of the input signal. For symmetric PDFs, the arithmetic mean will be the (abscissa) position of the symmetric axis. For example, a time domain audio signal usually has a mean value of approximately 0; if the signal has a DC offset, the mean will indicate the amount of the DC offset. When the PDF is not symmetric, then the calculation of the arithmetic mean is of limited use. In this case, the computation of the median (see Sect. 3.2.10) or the centroid (see Sect. 3.2.5) of the PDF might be more meaningful measures of the average.

3.2.2 Geometric Mean

The *geometric mean* is an average measure for sets of positive numbers that are ordered on a logarithmic scale. It can be computed with

$$M_x(0,n) = \sqrt[\mathcal{K}]{\prod_{i=i_s(n)}^{i_e(n)} x(i)} \tag{3.11}$$

$$= \exp\left(\frac{1}{\mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} \log\left[x(i)\right]\right). \tag{3.12}$$

Equation (3.12) is equivalent to Eq. (3.11) but avoids problems with computational accuracy for long blocks of data and large values at the computational cost of applying a logarithm to each signal value. The result of the geometric mean is a value between the minimum and maximum input signal value. The unit corresponds to the unit of the input signal.

3.2.3 Harmonic Mean

The *harmonic mean* is an average measure appropriate for averaging rates. It is

$$M_x(-1,n) = \frac{\mathcal{K}}{\sum_{i=i_s(n)}^{i_e(n)} 1/x(i)}. \tag{3.13}$$

3.2.4 Generalized Mean

A generalized expression for the calculation of different measures of mean is

$$M_x(\beta,n) = \sqrt[\beta]{\frac{1}{\mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} x^\beta(i)}. \tag{3.14}$$

Different values of β then lead to different measures:

- $\beta = 1$: arithmetic mean, (Sect. 3.2.1)
- $\beta = 2$: quadratic mean, or RMS (see Sect. 4.3.1)
- $\beta = -1$: harmonic mean (Sect. 3.2.3)
- $\beta \to 0$: geometric mean, (Sect. 3.2.2)
- $\beta \to -\infty$: minimum
- $\beta \to \infty$: maximum

3.2.5 Centroid

The *centroid* computes the *Center of Gravity (COG)* of a block of input values. It is computed by the index-weighted sum of the values divided by their unweighted sum:

$$v_{\mathrm{C}}(n) = \frac{\sum_{i=i_{\mathrm{s}}(n)}^{i_{\mathrm{e}}(n)} \big(i - i_{\mathrm{s}}(n)\big) \cdot x(i)}{\sum_{i=i_{\mathrm{s}}(n)}^{i_{\mathrm{e}}(n)} x(i)}. \tag{3.15}$$

The result will be a value in the range of $0 \leq v_{\mathrm{C}}(n) \leq \mathcal{K} - 1$. For more information, see Sect. 3.3.3 on the spectral centroid.

3.2.6 Variance and Standard Deviation

Both the *variance* and the *standard deviation* measure the spread of the input signal $x(i)$ around its arithmetic mean. The variance σ_x^2 is defined by

$$\sigma_x^2(n) = \frac{1}{\mathcal{K}} \sum_{i=i_{\mathrm{s}}(n)}^{i_{\mathrm{e}}(n)} \big(x(i) - \mu_x(n)\big)^2. \tag{3.16}$$

Strictly speaking this is the so-called *biased* estimate of the variance in contrast to the *unbiased* estimate

$$\sigma_{x,\mathrm{b}}^2(n) = \frac{1}{\mathcal{K} - 1} \sum_{i=i_{\mathrm{s}}(n)}^{i_{\mathrm{e}}(n)} \big(x(i) - \mu_x(n)\big)^2. \tag{3.17}$$

In case of $\mu_x(n) = 0$, the variance equals the power of the observed block of samples.

The standard deviation σ_x can be computed directly from the variance

$$\sigma_x(n) = \sqrt{\sigma_x^2(n)}. \tag{3.18}$$

In case of $\mu_x(n) = 0$, the standard deviation equals the RMS of the observed block of samples (see Sect. 4.3.1). The result of the standard deviation is in the range

$$0 \leq \sigma_x(n) \leq \max_{i \in [i_{\mathrm{s}}(n); i_{\mathrm{e}}(n)]} |x(i)|.$$

Table 3.1 Spectral skewness for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SSk}
silence	not def.
flat mag.	not def.
single peak (@ k_s)	not def.

The standard deviation is 0 for silent or constant input signals. Its unit corresponds to the input signal's unit.

Although the computation of standard deviation or variance of audio samples might be of interest in specific cases, it is more common in ACA to compute them from a series of features.

3.2.7 Skewness

The *skewness* is referred to as the third central moment of a variable divided by the cube of its standard deviation. It is defined by

$$v_{\mathrm{Sk}}(n) = \frac{1}{\sigma_x^3(n) \cdot \mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} \big(x(i) - \mu_x(n)\big)^3. \tag{3.19}$$

The skewness is a measure of the asymmetry of the PDF. It will be 0 for symmetric distributions, negative for distributions with their mass centered on the right (left-skewed), and positive for distributions with their mass centered on the left (right-skewed). Note that while every symmetric distribution has zero skewness the converse is not necessarily true. The range is unrestricted. It is not defined for signals with a standard deviation of 0.

3.2.7.1 Spectral Skewness

The *spectral skewness* measures the symmetry of the distribution of the spectral magnitude values around their arithmetic mean. It is defined by

$$v_{\mathrm{SSk}}(n) = \frac{2 \sum_{k=0}^{\mathcal{K}/2-1} \big(|X(k,n)| - \mu_{|X|}\big)^3}{\mathcal{K} \cdot \sigma_{|X|}^3}. \tag{3.20}$$

For all features computed from the spectrum we will present the feature output for three prototypical spectral shapes in a table; these shapes are

- *silence*: $|X(k,n)| = 0$
- *white noise* $|X(k,n)| = const$ and
- a *single spectral peak*

$$|X(k,n)| = 0|_{k \neq k_s} \vee |X(k,n)| = A|_{k=k_s}. \tag{3.21}$$

The spectral skewness is, unfortunately, not a good first example as it is not defined for either of these three signals (see Table 3.1).

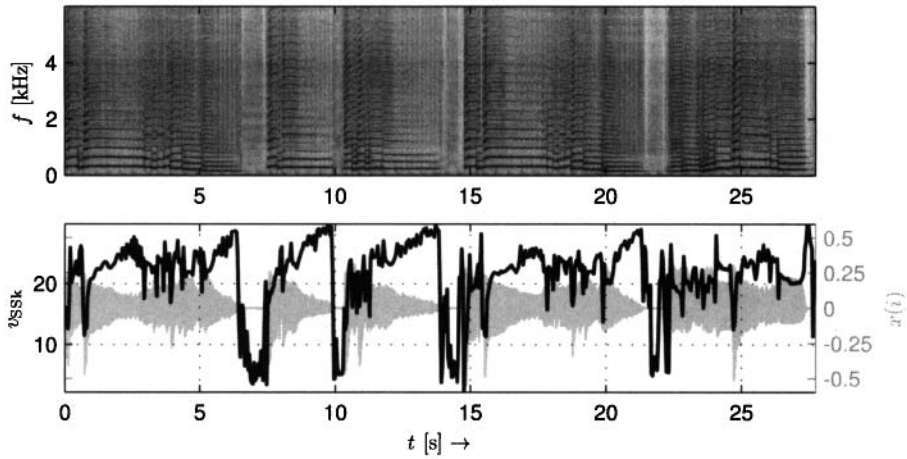

Figure 3.2 Spectrogram (top), waveform (bottom background), and spectral skewness (bottom foreground) of a saxophone signal

Figure 3.2 shows the spectral skewness for an example signal. During signal pauses the spectral skewness drops since the spectral magnitudes are similar, while at positions with high magnitudes at the fundamental frequency the magnitude spectrum is significantly skewed. The spectral skewness increases, for example, in the region between 12 s and 15 s due to the strong decrease of the higher harmonics compared to the lower harmonics.

3.2.8 Kurtosis

The *kurtosis* is referred to as the fourth central moment of a variable divided by the fourth power of the standard deviation:

$$v_K(n) = \frac{1}{\sigma_x^4(n) \cdot \mathcal{I}} \sum_{i=i_s(n)}^{i_e(n)} \Big(x(i) - \mu_x(n)\Big)^4 - 3. \tag{3.22}$$

The kurtosis is a measure of "non-Gaussianity" of the PDF; more specifically, it indicates the flatness (and peakiness, respectively) of the input values' distribution compared to the Gaussian distribution. It equals 0 for a Gaussian distribution (*mesokurtic*), is negative for a flatter distribution with a wider peak (*platykurtic*), and positive for distributions with a more acute peak (*leptokurtic*).

The range is not restricted, and similar to the skewness, the kurtosis is not defined for signals with a standard deviation of 0.

3.2.8.1 Spectral Kurtosis

The *spectral kurtosis* measures whether the distribution of the spectral magnitude values is shaped like a Gaussian distribution or not. It is defined by

$$v_{SK}(n) = \frac{2 \sum_{k=0}^{\mathcal{K}/2-1} \big(|X(k,n)| - \mu_{|X|}\big)^4}{\mathcal{K} \cdot \sigma_{|X|}^4} - 3. \tag{3.23}$$

Table 3.2 Spectral kurtosis for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SK}
silence	not def.
flat mag.	not def.
single peak (@ k_s)	not def.

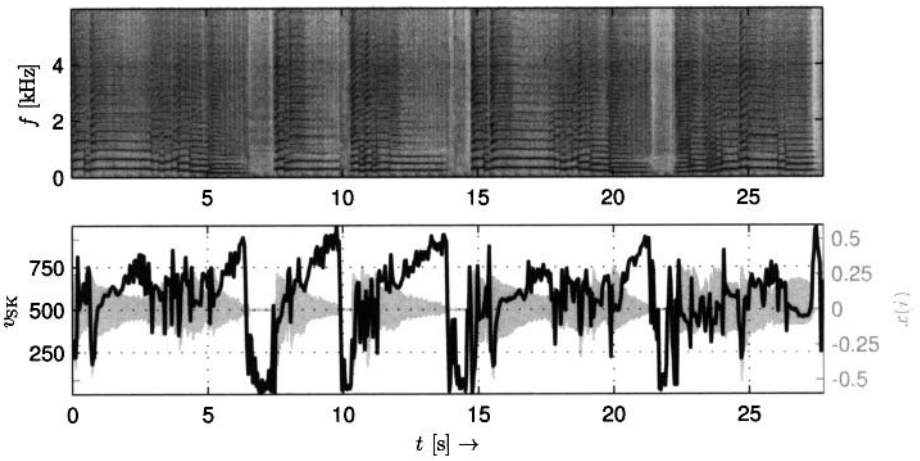

Figure 3.3 Spectrogram (top), waveform (bottom background), and spectral kurtosis (bottom foreground) of a saxophone signal

Table 3.2 shows that the spectral kurtosis is, similar to the spectral skewness, not defined for the three prototype signals.

Figure 3.3 shows the spectral kurtosis for a saxophone signal. While during the notes high values can be observed, indicating a very peaked distribution, the spectral kurtosis drops significantly during pauses.

3.2.9 Generalized Central Moments

The measures variance, skewness, and kurtosis are directly derived from the so-called central moments of different order. A central moment of order $\mathcal{O}$ is defined by

$$\gamma_{x,\beta}(n) = \sum_{i=i_{\mathrm{s}}(n)}^{i_{\mathrm{e}}(n)} \Big(x(i) - \mu_x(n)\Big)^{\mathcal{O}}. \tag{3.24}$$

3.2.10 Quantiles and Quantile Ranges

Quantiles can be computed from the PDF and can be used to divide the PDF into (equal sized) subsets. They are helpful in the description of asymmetric distributions or distributions with so-called outliers, occasional untypical values far from the median (see below).

If the PDF is divided into two quantiles, it is split into two parts each containing 50% of the overall number of observations. The position of the border between those two quantiles will be referred to as the *median* $Q_x(0.5)$:

$$Q_x(0.5) = x \Big|_{\int_{-\infty}^{x} p_x(y)\,dy = 0.5} \tag{3.25}$$

which equals the arithmetic mean in the case of symmetric distributions. Similarly, if the PDF is partitioned in four quantiles (so-called *quartiless*), the quantile borders would be $Q_x(0.25)$, $Q_x(0.5)$, and $Q_x(0.75)$.

In many cases, *quantile ranges* are of specific interest for the simplified description of a distribution's shape. The range spanned by 90% of the samples can be computed by $\Delta Q_x(0.9) = Q_x(0.95) - Q_x(0.05)$. This should be a good measure of the signal's range while discarding infrequent outliers, namely the upper and lower 5%.

The overall range of a signal is

$$\Delta Q_x(1.0) = \min\big(Q_x(1.0)\big) - \max\big(Q_x(0)\big). \tag{3.26}$$

3.3 Spectral Shape

Most of the features describing the spectral shape of an audio signal are closely related to the timbre of this signal. The *timbre* of a sound is referred to as its *sound color*, its *quality*, or its *texture*. Besides pitch and loudness, timbre is considered as "the third attribute of the subjective experience of musical tones" [37]. Timbre can be explained by two closely related phenomena, which will be referred to as *timbre quality* and *timbre identity*. The timbre quality allows humans to group together different sounds originating from the same source such as two recordings made with the same instrument. Timbre identity enables the differentiation of two sounds with the same tone characteristics (loudness, pitch if available) played on two instruments. Thus, the quality represents general timbre properties of a sound ("sounds like a violin"), while the timbre identity refers to instrument specifics ("one violin sounds different from the other").

Loudness and pitch are unidimensional properties, as sounds with different loudness or pitch can be ordered on a single scale from quiet to loud and low to high, respectively. Timbre is a multi-dimensional property [38, 39]; this complicates its definition. A good summary over the various attempts of the definition of the term timbre has been compiled by Sandell.[1] The most prominent example is probably the definition of the American Standards Association from 1960 that defined timbre as "that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar" [40]. This definition has been criticized repeatedly by researchers mainly because according to Bregman [41] it

- does not attempt to explain what timbre is, but only what timbre is *not*, i.e., loudness and pitch, and
- implies that timbre only exists for sounds with a pitch, implicating that, for example, percussive instruments do not have a timbre.

[1] Sandell, Greg: *Definitions of the word "Timbre"*. http://www.zainea.com/timbre.htm. Last retrieved on Nov. 16, 2011.

Early uses of the term *timbre* can be found in Blumenbach [42]. Helmholtz was probably the first to detect the dependency between the timbre of a sound and the relative amplitudes of the harmonics during the second half of the 19th century [43]. Although he noted that other influences play a role in defining the quality of a tone such as the "beginning" and "ending" of a sound, he restricted his definition of timbre ("Klangfarbe") to the harmonic amplitude distribution only.

Stumpf extended the definition of timbre by two more attributes [44]. He named the relative amplitude of harmonics, the form and length of the attack time and note endings, and additional sounds and noise as the third timbre-determining component.

Seashore restricted the term timbre during the first half of the 20th century to the harmonic structure that "is expressed in terms of the number, distribution, and relative intensity of its partials," but he additionally introduced the term *sonance*, referring to "the successive changes and fusions which take place within a tone from moment to moment" [45]. This distinction did, however, not find broad acceptance in the research community.

Nowadays, timbre is understood as the phenomenon that takes into account both spectral patterns and temporal patterns [39, 46]. Timbre perception is obviously influenced by numerous parameters of both the onset properties such as rise time, inharmonicities during the onset, etc. and numerous steady-state effects such as vibrato, tremolo, pitch instability, etc. [37]. In the following, we will restrict ourselves to measures of spectral shape since most of the features describing "temporal timbre" only work for individual monophonic notes as opposed to complex time-variant mixtures of signals. The presented features represent spectral shape and are technically motivated; therefore, there is not necessarily a direct relation to the human perception of timbre.

3.3.1 Spectral Rolloff

The *spectral rolloff* is a measure of the bandwidth of the analyzed block n of audio samples. The spectral rolloff $v_{\mathrm{SR}}(n)$ is defined as the frequency bin below which the accumulated magnitudes of the STFT $X(k,n)$ reach a certain percentage κ of the overall sum of magnitudes:

$$v_{\mathrm{SR}}(n) = i \left| \sum_{k=0}^{i} |X(k,n)| = \kappa \cdot \sum_{k=0}^{K/2-1} |X(k,n)| \right. \tag{3.27}$$

with common values for κ being 0.85 (85%) or 0.95 (95%).

The result of the spectral rolloff is a bin index in the range $0 \leq v_{\mathrm{SR}}(n) \leq K/2 - 1$. It can be converted either to Hz with Eq. (2.44) or to a parameter range between zero and one by dividing it by the STFT size $K/2 - 1$. Low results indicate insignificant magnitude components at high frequencies and thus a low audio bandwidth.

Table 3.3 shows the results for the spectral rolloff for three prototype spectral shapes. The behavior of the spectral rolloff at pauses in the input signal may require special consideration. While the result will equal zero for absolute silence, it may be quite large for noise, including pauses with low-level noise.

Table 3.3 Spectral rolloff for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SR}
silence	0
flat mag.	$\kappa \cdot \kappa/2 - 1$
single peak (@ k_s)	k_s

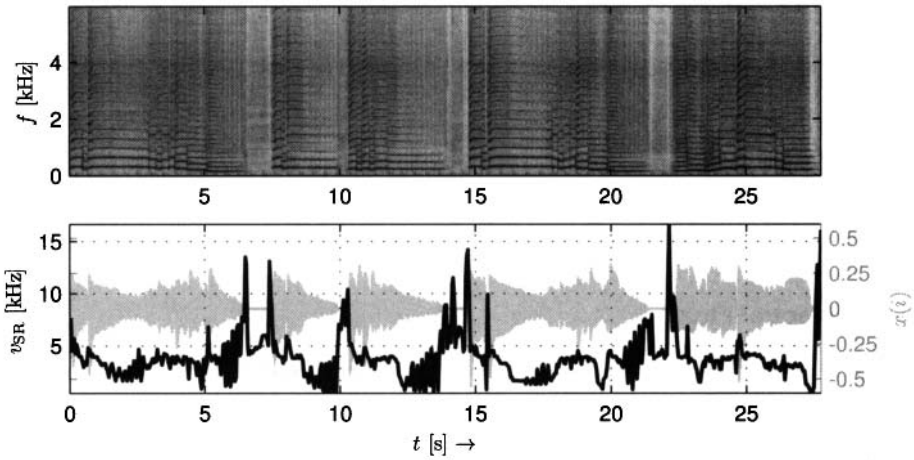

Figure 3.4 Spectrogram (top), waveform (bottom background), and spectral rolloff (bottom foreground) of a saxophone signal

Figure 3.4 shows the spectral rolloff for an example signal. It is comparably low in the presence of a tone and higher — although somewhat erratic — during the noise-filled pauses.

3.3.1.1 Common Variants

Spectral bins representing very low or very high frequencies may in many cases be considered to be unnecessary or unwanted for the analysis. Therefore, both sums in Eq. (3.27) may start and stop at pre-defined frequency boundaries $f_{\min}$, $f_{\max}$:

$$v_{\mathrm{SR},\Delta\mathrm{f}}(n) = i \left| \sum_{k=k(f_{\min})}^{i} |X(k,n)| = \kappa \cdot \sum_{k=k(f_{\min})}^{k(f_{\max})} |X(k,n)| \right| . \tag{3.28}$$

It is also common to use the power spectrum instead of the magnitude spectrum:

$$v_{\mathrm{SR},\mathrm{Pow}}(n) = i \left| \sum_{k=0}^{i} |X(k,n)|^2 = \kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k,n)|^2 \right| . \tag{3.29}$$

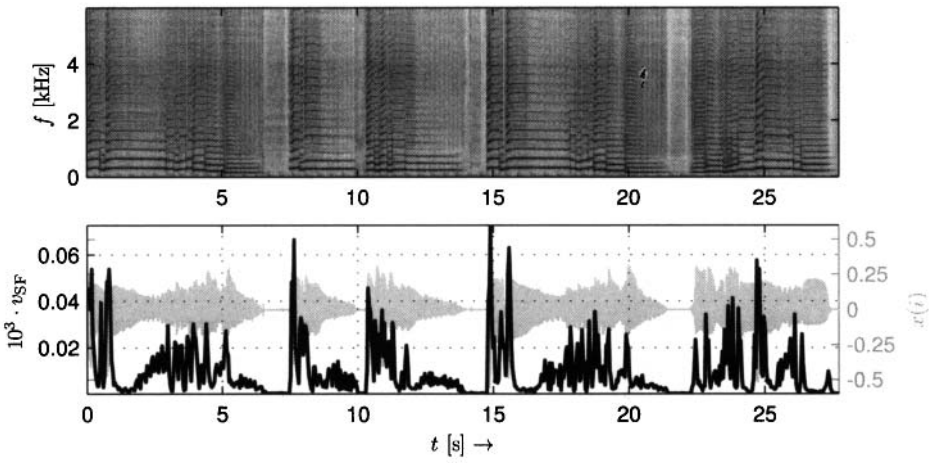

Figure 3.5 Spectrogram (top), waveform (bottom background), and spectral flux (bottom foreground) of a saxophone signal

3.3.2 Spectral Flux

The *spectral flux* measures the amount of change of the spectral shape. It is defined as the average difference between consecutive STFT frames:

$$v_{\mathrm{SF}}(n) = \frac{\sqrt{\sum_{k=0}^{K/2-1} \left(|X(k,n)| - |X(k,n-1)|\right)^2}}{K/2}. \tag{3.30}$$

The spectral flux can be interpreted as a rudimentary approximation to the sensation *roughness* which according to Zwicker and Fastl can be modeled as quasi-periodic change or a modulation in the excitation pattern levels [47].

The result of the spectral flux is a value within the range $0 \leq v_{\mathrm{SF}}(n) \leq A$ with A representing the maximum possible spectral magnitude. Thus, its output range depends on the normalization of the audio signal and the frequency transform. Low results indicate steady-state input signals or low input levels.

Figure 3.5 shows the spectral flux for an example signal. It is low during the stationary parts of the signal, such as during a note or a pause, and spikes at pitch changes and at the beginning of a new note.

3.3.2.1 Common Variants

The definition of the spectral flux above is the Euclidean distance of the two spectra as given in Eq. (5.59). The distance norm can also be generalized:

$$v_{\mathrm{SF}}(n,\beta) = \frac{\sqrt[\beta]{\sum_{k=0}^{K/2-1} \left(|X(k,n)| - |X(k,n-1)|\right)^\beta}}{K/2}. \tag{3.31}$$

Typical values for β range between $[0.25; 3]$, with $\beta = 1$ [manhattan distance, Eq. (5.60)] and $\beta = 2$ [Euclidean distance, Eq. (5.59)] being the most common.

In some applications such as note onset detection, only an increase in spectral energy is of interest; in these cases, the difference magnitude spectrum is computed[2]

$$\Delta X(k,n) = |X(k,n)| - |X(k,n-1)| \tag{3.32}$$

and all negative differences $\Delta X(k,n) < 0$ can be set to zero before summation while positive differences will be left unaltered. This is called *Half-Wave Rectification (HWR)*. Mathematically the HWR of a signal x is

$$\mathrm{HWR}(x) = \frac{x + |x|}{2}. \tag{3.33}$$

Alternative approaches can be used to derive a measure of spectral change. An example is the computation of the standard deviation of the difference magnitude spectrum $\Delta X(k,n)$:

$$v_{\mathrm{SF},\sigma}(n) = \sqrt{\frac{2}{K}\sum_{k=0}^{K/2-1}(\Delta X(k,n) - \mu_{\Delta X})^2}. \tag{3.34}$$

Another variant is to compute the logarithmic difference. This has the advantage of making the resulting feature independent of magnitude scaling but the disadvantage of zero-mean frames having to be handled individually:

$$v_{\mathrm{SF,log}}(n) = \frac{2}{K}\sum_{k=0}^{K/2-1}\log_2\left(\frac{|X(k,n)|}{|X(k,n-1)|}\right). \tag{3.35}$$

3.3.3 Spectral Centroid

The *spectral centroid* represents the COG of spectral energy (compare Sect. 3.2.5 for the time domain centroid). It is defined as the frequency-weighted sum of the power spectrum normalized by its unweighted sum:

$$v_{\mathrm{SC}}(n) = \frac{\sum_{k=0}^{K/2-1} k \cdot |X(k,n)|^2}{\sum_{k=0}^{K/2-1} |X(k,n)|^2}. \tag{3.36}$$

In the literature, numerous indications can be found that this position of energy concentration is well correlated with the timbre dimension *brightness* or *sharpness* [48–54].

The result of the spectral centroid is a bin index within the range $0 \leq v_{\mathrm{SC}}(n) \leq K/2-1$. It can be converted either to Hz by using Eq. (2.44) or to a parameter range between zero and one by dividing it by the STFT size $K/2 - 1$. Low results indicate significant low-frequency components and insignificant high frequency components and low "brightness."

Table 3.4 shows the results for the spectral centroid for three prototype spectral shapes. The behavior of the spectral centroid at pauses in the input signal requires special consideration as it is not defined for silence and will be comparably large for (low-level) noise.

Figure 3.6 shows the spectral centroid for an example signal. In the case of this monophonic signal one can see how the spectral centroid moves with the fundamental frequency during tonal parts, spikes at initial transients, and is high during pauses.

[2] Another distance measure can be used as well.

Table 3.4 Spectral centroid for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SC}
silence	not def.
flat mag.	$\frac{K/2-1}{2}$
single peak (@ k_s)	k_s

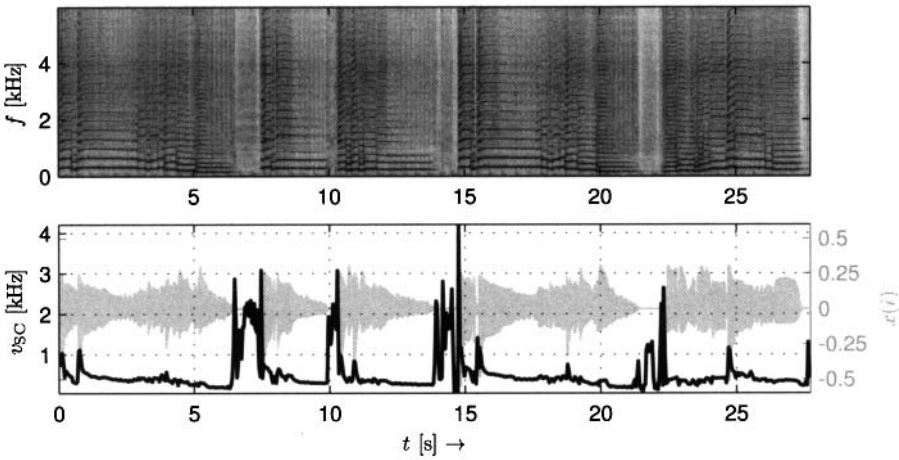

Figure 3.6 Spectrogram (top), waveform (bottom background), and spectral centroid (bottom foreground) of a saxophone signal

3.3.3.1 Common Variants

The magnitude spectrum may be used instead of the power spectrum:

$$v_{\mathrm{SC,m}}(n) = \frac{\sum_{k=0}^{K/2-1} k \cdot |X(k,n)|}{\sum_{k=0}^{N/2-1} |X(k,n)|}. \tag{3.37}$$

Zwicker and Fastl presented a psycho-acoustic model of sharpness that uses the excitation patterns to compute the sharpness [47]. It differs from Eq. (3.36) mainly in two points. First, it is computed on a non-linear bark scale, namely the so-called critical band rate which models the non-linearity of human frequency perception (compare Sect. 5.1.1.2). Second, it utilizes a psycho-acoustic loudness measure instead of the spectral power; this loudness model takes into account masking and other perceptual effects. While ignoring the difference between loudness and power, the idea of a "more human" non-linear frequency scale has been adapted for the definition of the spectral centroid in the MPEG-7 standard [33]. The critical band rate is approximated by applying a logarithm to the

Table 3.5 Spectral spread for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SS}
silence	not def.
flat mag.	$4/\kappa \cdot \sum_{k=0}^{\kappa/4-1} (\kappa/4 - k)^2$
single peak (@ k_s)	0

frequencies with a reference point of $f_{ref} = 1000\,\text{Hz}$:

$$v_{SC,log}(n) = \frac{\sum_{k=k(f_{min})}^{\kappa/2-1} \log_2\left(\frac{f(k)}{f_{ref}}\right) \cdot |X(k,n)|^2}{\sum_{k=k(f_{min})}^{N/2-1} |X(k,n)|^2}. \tag{3.38}$$

In this specific MPEG-definition, all bins corresponding to frequencies below 62.5 Hz are combined to one band with a mid-frequency of 31.25 Hz.

3.3.4 Spectral Spread

The *spectral spread*, sometimes also referred to as *instantaneous bandwidth*, describes the concentration of the power spectrum around the spectral centroid and is a rather technical description of spectral shape. It can be interpreted as the standard deviation of the power spectrum around the spectral centroid. Its definition is

$$v_{SS}(n) = \sqrt{\frac{\sum_{k=0}^{\kappa/2-1} (k - v_{SC}(n))^2 \cdot |X(k,n)|^2}{\sum_{k=0}^{\kappa/2-1} |X(k,n)|^2}}. \tag{3.39}$$

There are indications that the spectral spread is of some relevance in describing the perceptual dimensions of timbre [52].

The result of the spectral spread is a bin range of $0 \leq v_{SS}(n) \leq \kappa/4$. It can be converted either to Hz by using Eq. (2.44) or to a parameter range between zero and one by dividing it by $\kappa/4$. Low results indicate the concentration of the spectral energy at a specific frequency region. As the spectral centroid, the spectral spread is not defined for audio blocks with no spectral energy (silence) and will result in high values if the input signal contains (low-level) white noise.

Table 3.5 shows the results for the spectral spread for three prototype spectral shapes. Figure 3.7 shows the spectral spread for an example signal. Most prominent are the high feature values during pauses and at transients; the spectral spread is low during notes for this monophonic signal. When the higher harmonics slowly disappear between 12 and 15 s, the spread of the signal decreases accordingly.

3.3.4.1 Common Variants

The definition of the spectral spread has to conform with the definition of the spectral centroid. If the spectral centroid has been calculated from the magnitude spectrum instead

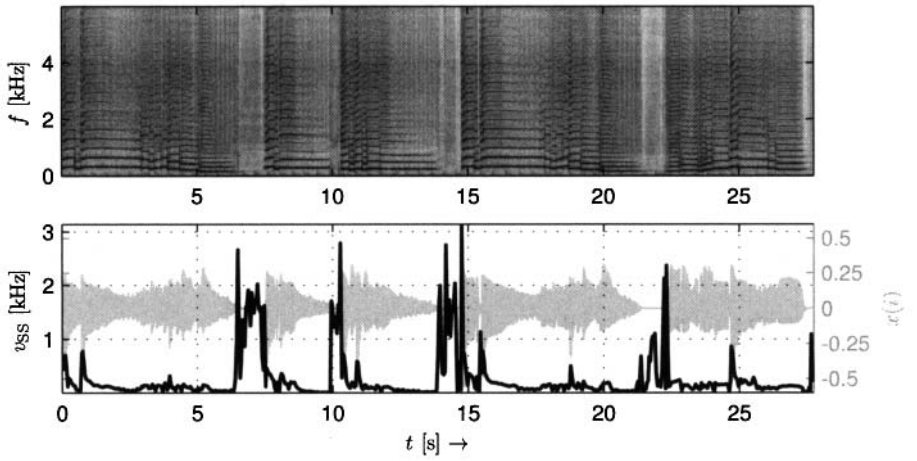

Figure 3.7 Spectrogram (top), waveform (bottom background), and spectral spread (bottom foreground) of a saxophone signal

of the power spectrum, then the spectral spread should use the magnitude spectrum as well. In the case of the MPEG-7 definition, a logarithmic frequency scale has to be used for the calculation of the spectral spread as well:

$$v_{\mathrm{SS,log}}(n) = \sqrt{\frac{\sum_{k=k(f_{\min})}^{\kappa/2-1} \left(\log_2\left(\frac{f(k)}{1000\,\mathrm{Hz}}\right) - v_{\mathrm{SC}}(n)\right)^2 \cdot |X(k,n)|^2}{\sum_{k=k(f_{\min})}^{\kappa/2-1} |X(k,n)|^2}}. \tag{3.40}$$

3.3.5 Spectral Decrease

The *spectral decrease* estimates the steepness of the decrease of the spectral envelope over frequency. It is defined by [34]

$$v_{\mathrm{SD}}(n) = \frac{\sum_{k=1}^{\kappa/2-1} \frac{1}{k} \cdot \left(|X(k,n)| - |X(0,n)|\right)}{\sum_{k=1}^{\kappa/2-1} |X(k,n)|}. \tag{3.41}$$

The result of the spectral decrease is a value $v_{\mathrm{SD}}(n) \leq 1$. Low results indicate the concentration of the spectral energy at bin 0. The spectral decrease is not defined for audio blocks with no spectral energy (silence).

Table 3.6 Spectral decrease for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{SD}
silence	not def.
flat mag.	0
single peak (@ k_s)	$1/k_s$

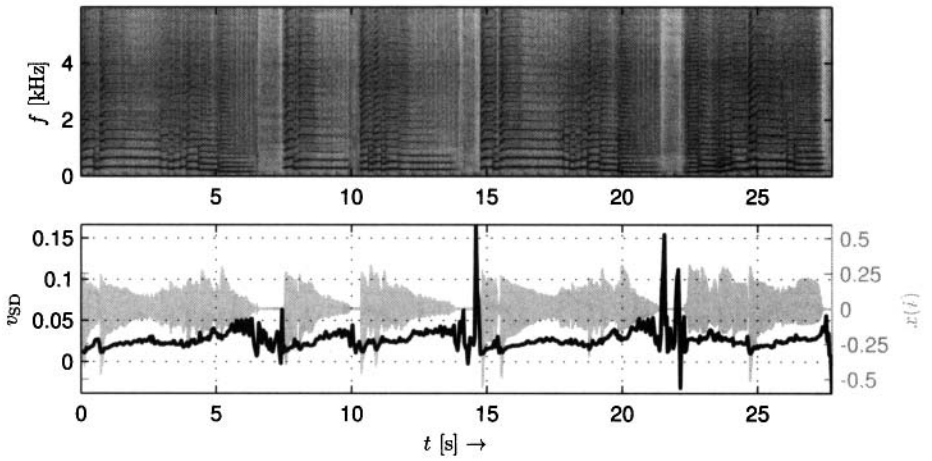

Figure 3.8 Spectrogram (top), waveform (bottom background), and spectral decrease (bottom foreground) of a saxophone signal

Table 3.6 shows the results for the spectral decrease for three prototype spectral shapes.

Figure 3.8 shows the spectral decrease for an example signal. It is difficult to draw any conclusions from the graph except that the feature behaves erratically during the pauses.

3.3.5.1 Common Variants

Reducing the spectral analysis range might lead to more meaningful results in some cases. This can be done by using a lower and upper bound k_{l} and k_{u}, respectively:

$$v_{\mathrm{SD}}(n) = \frac{\sum_{k=k_{\mathrm{l}}}^{k_{\mathrm{u}}} \frac{1}{k} \cdot \left(|X(k,n)| - |X(k_{\mathrm{l}}-1,n)|\right)}{\sum_{k=k_{\mathrm{l}}}^{k_{\mathrm{u}}} |X(k,n)|}. \tag{3.42}$$

3.3.6 Spectral Slope

The *spectral slope* is — similar to the spectral decrease — a measure of the slope of the spectral shape. It is calculated using a linear approximation of the magnitude spectrum; more specifically, a linear regression approach is used. In the presented form, the linear function is modeled from the magnitude spectrum. Its slope is then estimated with the equation

Table 3.7 Spectral slope for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude, the magnitude at the bin k_s equals A)

Spectral Shape	v_{SSl}
silence	0
flat mag.	0
single peak (@ k_s)	$\frac{(k_s - \mathcal{K}/4) \cdot (A - 2A/\mathcal{K})}{\sum_{k=0}^{\mathcal{K}/2-1}(k - \mu_k)^2}$

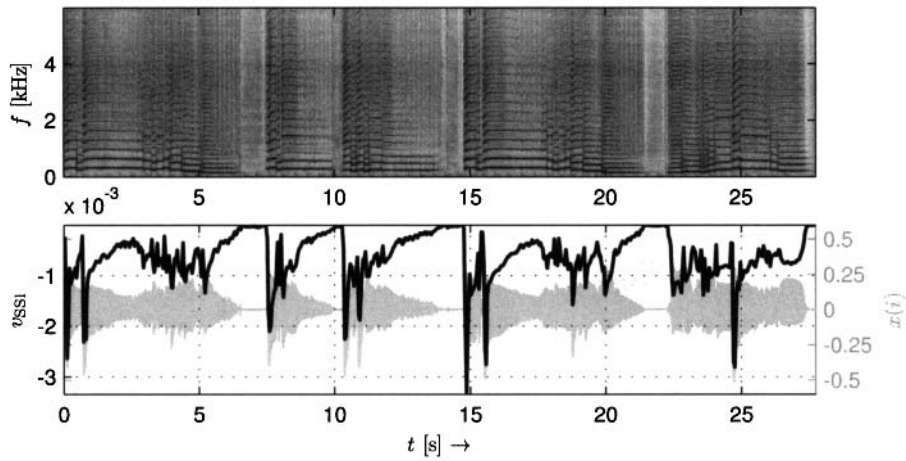

Figure 3.9 Spectrogram (top), waveform (bottom background), and spectral slope (bottom foreground) of a saxophone signal

$$v_{\mathrm{SSl}}(n) = \frac{\sum_{k=0}^{\mathcal{K}/2-1}(k - \mu_k)(|X(k,n)| - \mu_{|X|})}{\sum_{k=0}^{\mathcal{K}/2-1}(k - \mu_k)^2} \tag{3.43}$$

$$= \frac{\mathcal{K}\sum_{k=0}^{\mathcal{K}/2-1} k \cdot |X(k,n)| - \sum_{k=0}^{\mathcal{K}/2-1} k \cdot \sum_{k=0}^{\mathcal{K}/2-1} |X(k,n)|}{\mathcal{K} \cdot \sum_{k=0}^{\mathcal{K}/2-1} k^2 - \left(\sum_{k=0}^{\mathcal{K}/2-1} k\right)^2}. \tag{3.44}$$

The result of the spectral slope depends on the amplitude range of the spectral magnitudes.

Table 3.7 shows the results for the spectral slope for three prototype spectral shapes.

Figure 3.9 shows the spectral slope for an example signal. It is maximal for the noisy pauses and increases with disappearing higher harmonics.

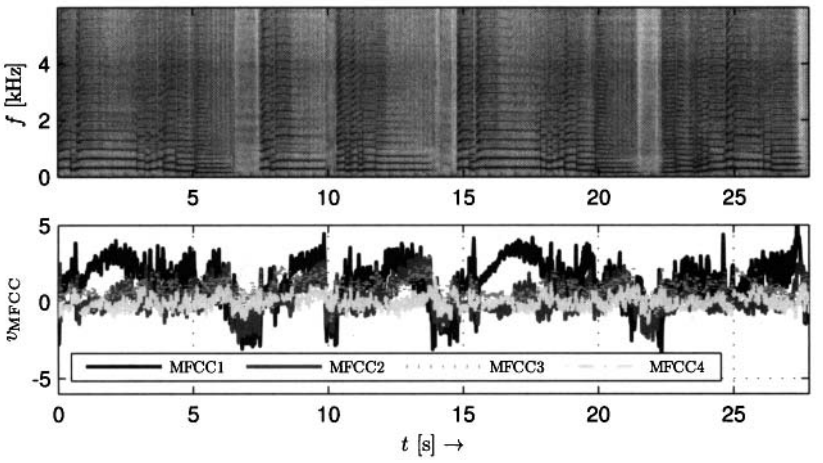

Figure 3.10 Spectrogram (top) and mel frequency cepstral coefficients 1–4 (bottom) of a saxophone signal

3.3.7 Mel Frequency Cepstral Coefficients

The *Mel Frequency Cepstral Coefficients (MFCCs)* can be seen as a compact description of the shape of the spectral envelope of an audio signal. The jth coefficient $v^j_{\mathrm{MFCC}}(n)$ can be calculated with

$$v^j_{\mathrm{MFCC}}(n) = \sum_{k'=1}^{\mathcal{K}'} \log\left(|X'(k',n)|\right) \cdot \cos\left(j \cdot \left(k' - \frac{1}{2}\right)\frac{\pi}{\mathcal{K}'}\right) \tag{3.45}$$

with $|X'(k',n)|$ being the mel-warped magnitude spectrum at the signal block. The calculation is based on the following steps:

1. computation of the mel-warped (see Sect. 5.1.1.1) spectrum with a bank of overlapping band-pass filters,
2. taking the logarithm of the magnitude of each resulting band, and
3. calculating the *Discrete Cosine Transform (DCT)* on the resulting bands. The DCT equals the real (cosine) part of an FT.

The MFCCs have been widely used in the field of speech signal processing since their introduction in 1980 [55] and have been found to be useful in music signal processing applications as well [56–59]. In the context of audio signal classification, it has been shown that a small subset of the resulting MFCCs as shown in Fig. 3.10 already contains the principal information [60, 61] — in most cases the number of used MFCCs varies in the range from 4 to 20.

The calculation is closely related to the calculation of the cepstrum as introduced in Sect. 5.3.3.7 as it transforms a logarithmic spectral representation. The main difference to the standard cepstrum is the use of a non-linear frequency scale (the mel scale) to model the non-linear human perception of pitch and the use of the DCT instead of a DFT. The mel-warped basis functions for the DCT are displayed in Fig. 3.11.

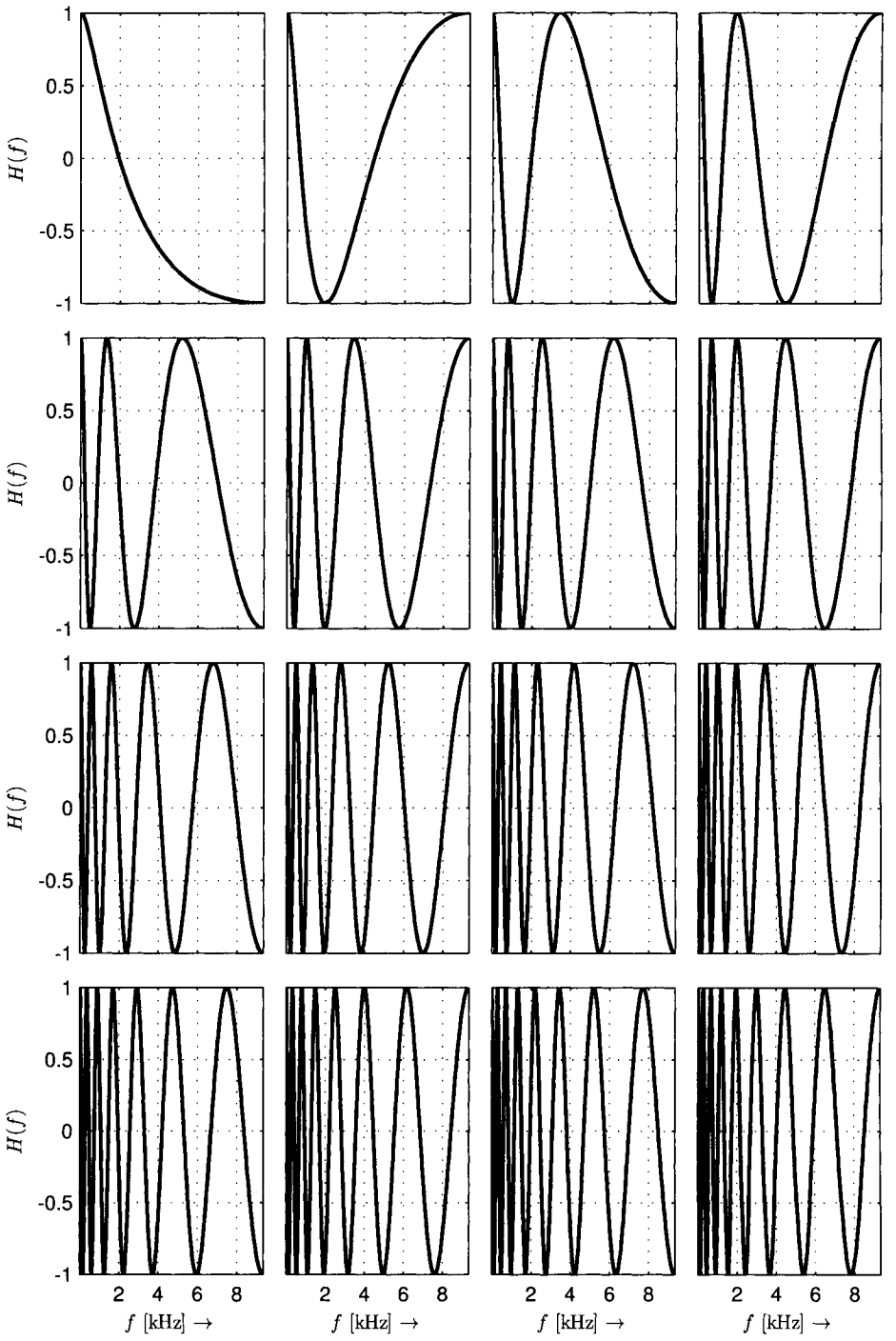

Figure 3.11 Warped cosine-shaped transformation basis functions for the computation of MFCC coefficients (order increases from left to right and from top to bottom)

Table 3.8 Properties of three popular MFCC implementations

Property	*DM*	*HTK*	*SAT*
Num. filters	20	24	40
Mel scale	lin/log	log	lin/log
Freq. range	[100; 4000]	[100; 4000]	[200; 6400]
Normalization	Equal height	Equal height	Equal area

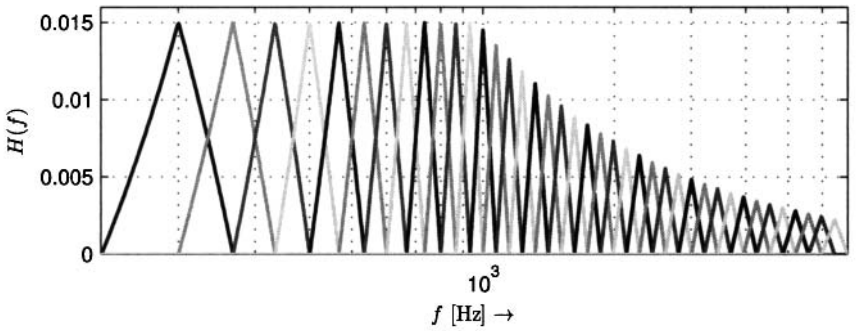

Figure 3.12 Magnitude transfer function of the filterbank for MFCC computation as used in Slaney's *Auditory Toolbox*

The mel warping of the spectrum frequently leads to the conclusion that the MFCCs are a "perceptual" feature. This is only partly true as there is no psycho-acoustic evidence to motivate the application of the DCT. Also, there is no direct correlation between the MFCCs and known perceptual dimensions.

The result of the MFCCs depends on the amplitude range of the spectral power. The zeroth MFCC $v^0_{\mathrm{MFCC}}(n)$ is usually ignored as it has no relevance in describing the timbre. It is simply a scaled measure of the energy in decibel. The MFCCs are not defined for silence as input signal.

The first four coefficients are shown in Fig. 3.10. Despite their proven usefulness it is difficult to identify non-trivial relationships to the input signal.

3.3.7.1 Common Variants

The differences between MFCC implementations can be found mainly in the computation of the mel-warped spectrum, i.e., in number, spacing, and normalization of the filters. Table 3.8 shows the differences between the three most popular MFCC implementations, the original introduced by Davis and Mermelstein (DM) [55], the implementation in the *HMM Toolkit (HTK)* software [62], and the implementation in Slaney's *Auditory Toolbox* (SAT) [27].

Figure 3.12 shows the triangular filter shapes as used in the Slaney's *Auditory Toolbox*.

Section 5.1.1.1 lists the typical mel scale models used for the non-linear frequency warping. It is also possible to use other filter shapes or to compute MFCCs directly from the power spectrum by using warped cosine basis functions as shown in Fig. 3.11 [63]. The power spectrum might also be approximated by other means such as through linear prediction coefficients.

Table 3.9 Spectral crest factor for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{Tsc}
silence	not def.
flat mag.	$2/\kappa$
single peak (@ k_s)	1

3.4 Signal Properties

3.4.1 Tonalness

Measures of *tonalness* estimate the amount of *tonal* components in the signal as opposed to noisy components. Tonalness is thus a measure related to sound quality. The somewhat unusual term *tonalness* is used here to distinguish this measure from the musical term *tonality* which describes a specific harmonic or key context. No specific measure of tonalness has itself established as de-facto standard, meaning that there are various different approaches to measure the tonalness of a signal. They have in common that for a signal considered to be tonal, they expect a high amount of periodicity and a low amount of noisy components. In that sense, the most tonal signal is a sinusoidal signal, and the most non-tonal signal is (white) noise. As an alternative to measuring the tonalness one could also find a feature for the noisiness which would be an "inverse" tonalness measure.

3.4.1.1 Spectral Crest Factor

A very simple measure of tonalness compares the maximum of the magnitude spectrum with the sum of this magnitude spectrum, a measure which will be referred to as *spectral crest factor*. It is defined by

$$v_{\mathrm{Tsc}}(n) = \frac{\max\limits_{0 \le k \le \kappa/2-1} |X(k,n)|}{\sum\limits_{k=0}^{\kappa/2-1} |X(k,n)|}. \tag{3.46}$$

The result of the spectral crest factor is a value between $2/\kappa \le v_{\mathrm{Tsc}}(n) \le 1$. Low results indicate a flat magnitude spectrum and high results indicate a sinusoidal. The spectral crest factor is not defined for audio blocks with no spectral energy (silence).

Table 3.9 shows the results for the spectral crest factor for three prototype spectral shapes.

Figure 3.13 shows the spectral crest factor for an example signal. It is low for noisy parts during the pauses and higher during the tonal passages. With decreasing amplitude of higher harmonics the spectral crest factor increases as the spectral energy is more and more concentrated at a single spectral bin.

Common Variants

A common variant is to replace the sum in the denominator by the arithmetic mean of the magnitude spectrum. This scales the range of the spectral crest factor.

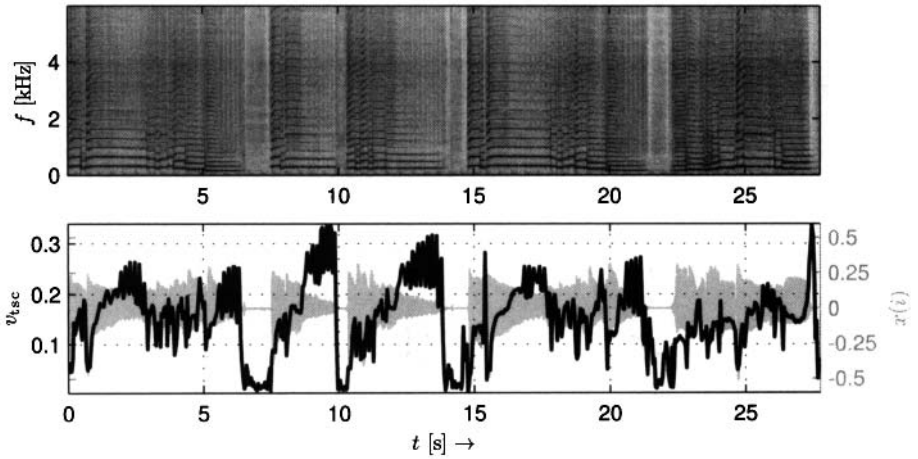

Figure 3.13 Spectrogram (top), waveform (bottom background), and tonalness feature spectral crest factor (bottom foreground) of a saxophone signal

Table 3.10 Spectral flatness for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{Tf}
silence	not def.
flat mag.	1
single peak (@ k_s)	0

3.4.1.2 Spectral Flatness

The *spectral flatness* is the ratio of geometric mean and arithmetic mean of the magnitude spectrum. It is defined by [64]

$$v_{\mathrm{Tf}}(n) = \frac{\sqrt[\kappa/2]{\prod_{k=0}^{\kappa/2-1} |X(k,n)|}}{2/\kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k,n)|} = \frac{\exp\left(2/\kappa \cdot \sum_{k=0}^{\kappa/2-1} \log\left(|X(k,n)|\right)\right)}{2/\kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k,n)|}. \tag{3.47}$$

The latter formulation uses the arithmetic mean of the logarithmic magnitude spectrum in the numerator in order to avoid problems with computing accuracy.

The result of the spectral flatness is a value larger than 0. The upper limit depends on the maximum spectral magnitude. Low results hint toward a non-flat — possibly a tonal — spectrum, while high results indicate a flat (or noisy) spectrum. The spectral flatness is thus a measure of noisiness as opposed to tonalness. However, as soon as only the magnitude at one individual bin equals 0, v_{Tf} will be zero as well.

Table 3.10 shows the results for the spectral flatness for three prototype spectral shapes. The behavior of the spectral flatness at pauses in the input signal requires special consideration as it is not defined for silence and will be comparably large for (low-level) noise.

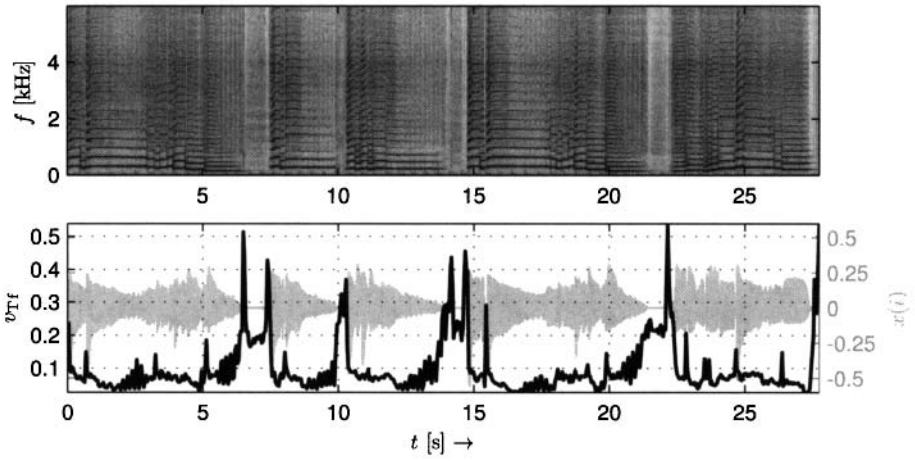

Figure 3.14 Spectrogram (top), waveform (bottom background), and tonalness feature spectral flatness (bottom foreground) of a saxophone signal

Figure 3.14 shows the spectral flatness for an example signal. It is low during tonal passages, high in noisy pauses, and produces spikes at transients.

Common Variants

It is common to use the power spectrum instead of the magnitude spectrum in order to emphasize peaks.

To avoid problems with individual zero magnitudes having too large an impact on the overall result, the magnitude spectrum can be smoothed. One typical approach is to compute the arithmetic mean of a group of neighboring spectral coefficients, which is basically the same as applying an MA filter to the magnitude spectrum. However, the length of the filter might also increase with frequency to take into account the lower frequency resolution of the human ear at higher frequencies.

In many cases, more useful information can be gathered if the spectral flatness calculation takes only magnitudes within a pre-defined frequency range into account, as opposed to computing it from the whole spectrum. The MPEG-7 standard recommends a frequency range of from 250 Hz to 16 kHz, divided into 24 slightly overlapping frequency bands with quarter-octave bandwidth [33]. Since the spectral flatness is then computed for each individual frequency band, the result per STFT is a vector of spectral flatness results.

3.4.1.3 Tonal Power Ratio

A straightforward way of computing the tonalness of a signal is to compute the ratio of the tonal power $E_{\mathrm{T}}(n)$ to the overall power:

$$v_{\mathrm{Tpr}} = \frac{E_{\mathrm{T}}(n)}{\sum_{i=0}^{K/2-1} |X(k,n)|^2}. \tag{3.48}$$

The interesting part in this case is naturally the estimation of the power of the tonal components. An approach to detecting tonal components in the spectrum is outlined in

Table 3.11 Tonal power ratio for the three prototypical spectral shapes *silence* (zero magnitude at all bins), *flat* (same amplitude at all bins), and *peak* (all bins except one have zero magnitude)

Spectral Shape	v_{Tpr}
silence	not def.
flat mag.	0
single peak (@ k_s)	1

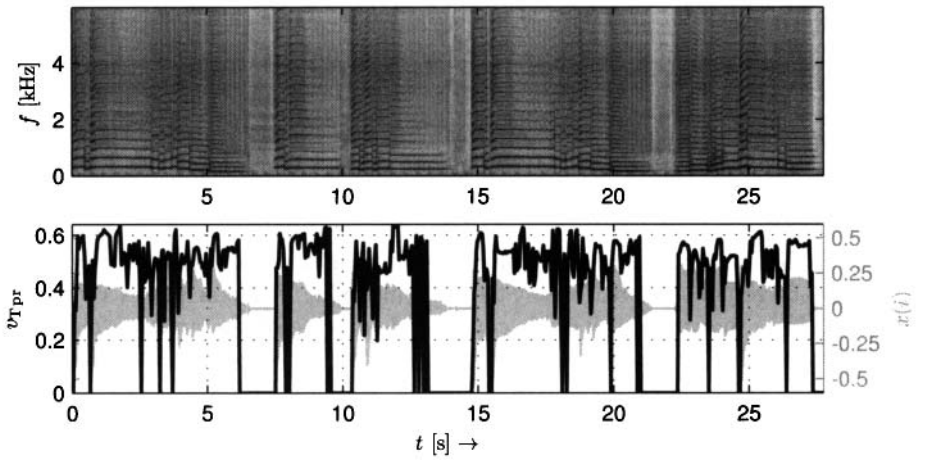

Figure 3.15 Spectrogram (top), waveform (bottom background), and tonalness feature tonal power ratio (bottom foreground) of a saxophone signal

Sect. 5.3.2.2. A simpler approximation to estimating the tonal energy is summing all bins k which

- are a local maximum: $|X(k-1,n)|^2 \leq |X(k,n)|^2 \geq |X(k+1,n)|^2$ and
- lie above a threshold G_T.

The result of the tonal power ratio is a value between $0 \leq v_{\mathrm{Tpr}} \leq 1$. Low results hint toward a flat (noisy) spectrum or a block with low input level while high results indicate a tonal spectrum.

Table 3.11 shows the results for the tonal power ratio for three prototype spectral shapes. The behavior of the spectral flatness at pauses in the input signal requires special consideration as it is not defined for silence and will be comparably large for (low-level) noise.

Figure 3.15 shows the tonal power ratio for an example signal. It is zero in noisy pauses, high for tonal passages and, in the case of this simple saxophone signal, drops distinctively at the initial transients at note beginnings.

3.4.1.4 Maximum of Autocorrelation Function

The ACF of a time signal yields local maxima where the ACF lag matches the wavelengths of the signal-inherent periodicities (see Sect. 2.2.6.2). The less periodic and therefore less tonal the signal is, the lower is the value of such maxima. The absolute value of the overall *ACF maximum* is therefore a simple estimate of the signal's tonalness:

$$v_{\mathrm{Ta}}(n) = \max_{0 \leq \eta \leq K-1} |r_{xx}(\eta, n)|. \tag{3.49}$$

Values in the main lobe of the ACF around lag $\eta = 0$ have to be discarded to ensure more reliable results. Different approaches can be used to ignore the main lobe:

- *Minimum lag*: Assuming that a maximum of interest will not be found at high frequencies (small lags and period lengths, respectively), the search for the maximum can be started at a pre-defined lag, ignoring values at smaller lags. The lower the expected maximum frequency is, the larger can the minimum lag can be. Depending on the task at hand and the sample rate, the maximum frequency might be too high to correspond to a reasonably large minimum lag. For example, at a sample rate of 48 kHz a frequency of 9.6 kHz corresponds to a lag of 5 samples, a frequency of 4.8 kHz to a lag of 10 samples, and a frequency of 1920 Hz to a lag of 25 samples.

- *Minimum magnitude threshold*: Maxima are only detected at lags larger than the lag η_{r}. This lag is the smallest lag at which r_{xx} crosses a pre-defined threshold G_{r}

$$\eta_{\mathrm{r}} = \operatorname*{argmin}_{0 \leq \eta \leq K-1} \left(r_{xx}(\eta) < G_{\mathrm{r}} \right). \tag{3.50}$$

  Theoretically, however, the threshold might never be crossed; this case has to be considered in the implementation.

- *Search range from the first local minimum*: Only consider maxima at lags larger than the lag of the "first" local minimum. The idea is to avoid the detection of "insignificant" local maxima in the main lobe around lag $\eta = 0$, but depending on the signal, a local minimum might be detected at a very low lag.

The best solution is to combine these approaches and possibly to find additional ways fitted to the problem to ensure meaningful results.

The result is a value between $0 \leq v_{\mathrm{Ta}}(n) \leq 1$. This ACF-based feature will work more reliable for monophonic signals or signals with a limited number of fundamental frequencies. Low results indicate a non-periodic signal and high results a periodic signal.

Table 3.12 Tonalness feature acf maximum for the three prototypical signal types *silence* (zero magnitude at all samples), *white noise* and a sinusoidal signal

Spectral Shape	v_{Ta}
silence	0
white noise	0
sinusoidal	1

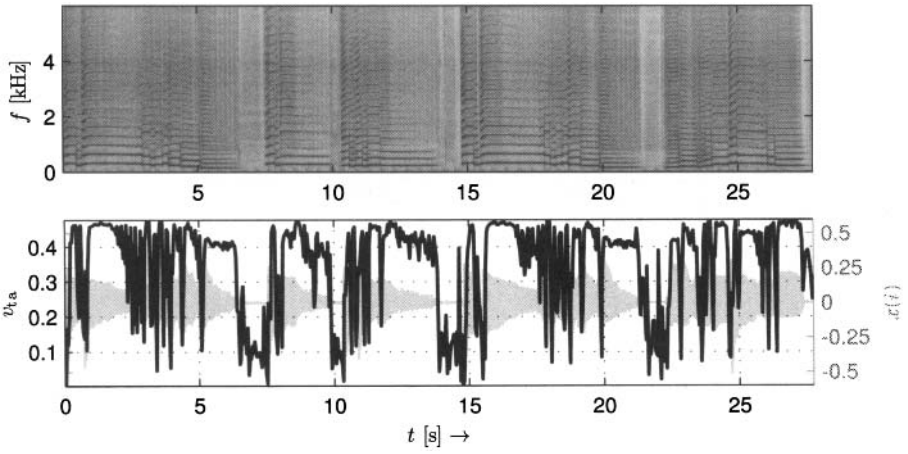

Figure 3.16 Spectrogram (top), waveform (bottom background), and tonalness feature ACF maximum (bottom foreground) of a saxophone signal

Table 3.12 shows the results for the ACF maximum for three prototype signals.

Figure 3.16 shows the ACF maximum for an example signal. As expected, there is the tendency of giving low values at noisy pause segments and higher values for tonal segments.

3.4.1.5 Predictivity Ratio

The *predictivity ratio* is a measure of how well the audio signal can be predicted by $\mathcal{O}$-order linear prediction (see Sect. 2.2.7). Each sample i is predicted using the preceding sample values and the prediction coefficients b_j:

$$\hat{x}(i) = \sum_{j=1}^{\mathcal{O}} b_j \cdot x(i-l). \tag{3.51}$$

The less noisy the signal is, the smaller the error e_{P} between the original and the predicted signal will be. Periodic and thus tonal signals will yield small prediction errors while noisy signals will result in high prediction errors. The power of the prediction error is therefore a measure of tonalness or more precisely a measure of noisiness as it will approach 0 for

Table 3.13 Tonalness feature predictivity ratio for the three prototypical signal types *silence* (zero magnitude at all samples), *white noise*, and a sinusoidal signal

Spectral Shape	v_{Tp}
silence	not def.
white noise	high
sinusoidal	$\rightarrow 0$

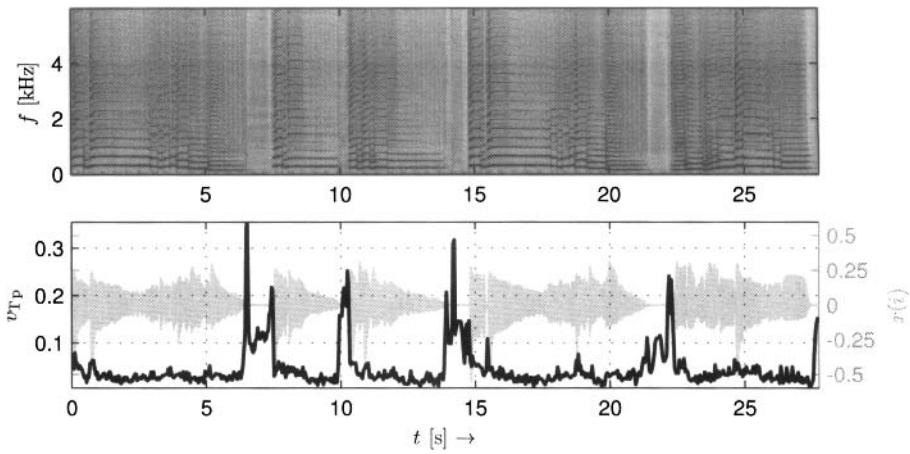

Figure 3.17 Spectrogram (top), waveform (bottom background), and tonalness feature predictivity ratio (bottom foreground) of a saxophone signal

tonal signals:

$$v_{\mathrm{Tp}}(n) = \sqrt{\frac{\sum_{i=i_s(n)}^{i_e(n)} \left(x(i) - \hat{x}(i)\right)^2}{\sum_{i=i_s(n)}^{i_e(n)} x^2(k)}}. \tag{3.52}$$

The result is a value larger or equal to 0. Low results indicate a periodic signal and high results a non-periodic signal.

Table 3.13 shows the results for the predictivity ratio for three prototype signals.

Figure 3.17 shows the predictivity ratio for an example signal with a predictor length of 12 coefficients. It clearly separates the noisy pause segments (high values) from the tonal segments (low values); the individual tonal parts cannot be distinguished with the feature.

3.4.1.6 Spectral Predictivity

The *spectral predictivity* is a measure of tonalness computed from overlapping STFTs. The magnitude and phase of each spectral bin are predicted with a simple predictor with fixed coefficients:

$$|\hat{X}(k,n)| = 2 \cdot |X(k,n-1)| - |X(k,n-2)|, \tag{3.53}$$

$$\hat{\phi}_X(k,n) = 2 \cdot \Phi_X(k,n-1) - \Phi_X(k,n-2). \tag{3.54}$$

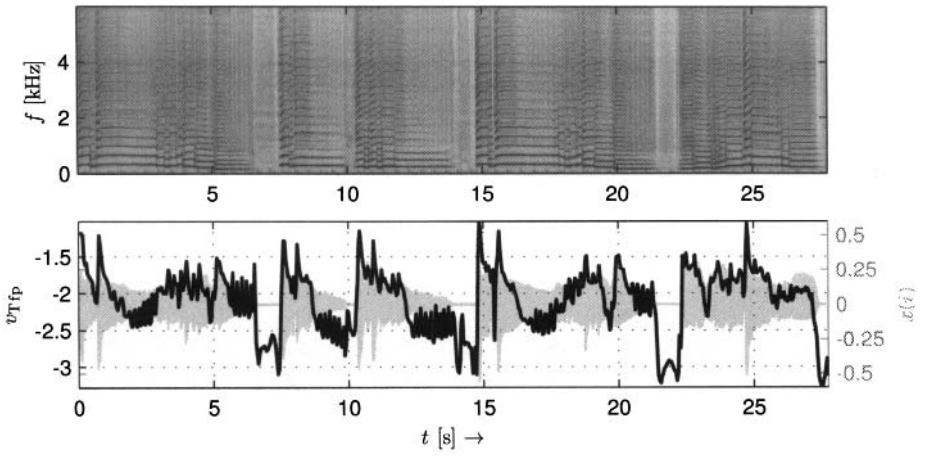

Figure 3.18 Spectrogram (top), waveform (bottom background), and tonalness feature spectral predictivity (bottom foreground) of a saxophone signal

The prediction error $e_{\mathrm{Tfp}}(k,n)$ is then defined by

$$e_{\mathrm{Tfp}}(k,n) = \left| \frac{|X(k,n)|e^{\Phi_X(k,n)} - |\hat{X}(k,n)|e^{\hat{\phi}_X}(k,n)}{|X(k,n)| \cdot |\hat{X}(k,n)|} \right| \tag{3.55}$$

and the resulting tonalness

$$v_{\mathrm{Tfp}}(n) = \frac{c_1 + c_2 \cdot \sum_{k=0}^{K/2-1} \log\left(e_{\mathrm{Tfp}}(k,n)\right)}{K/2} \tag{3.56}$$

with c_1 and c_2 as constants to be arbitrarily selected (MPEG: $c_1 = -0.299$, $c_2 = -0.43$).

The spectral predictivity as shown in Fig. 3.18 is used in the psycho-acoustic model II of the audio coding standards by the *Motion Picture Experts Group (MPEG)* [65].

3.4.2 Autocorrelation Coefficients

Infrequently, the first *autocorrelation coefficients* are used directly to describe statistical properties of the signal

$$v_{\mathrm{ACF}}^{\eta}(n) = r_{xx}(\eta, n) \quad \text{with } \eta = 1, 2, 3, \ldots \tag{3.57}$$

with $r_{xx}(\eta, n)$ being the ACF as defined in Sect. 2.2.6.2.

The number of used coefficients usually varies between 6 and 64 depending on requirements and sample rate. Each coefficient is in the range of $-1 \leq r_{xx}(\eta, n) \leq 1$. The faster the coefficients decrease with increasing lag, the "whiter" the signal can be assumed to be.

Table 3.14 shows typical results for the autocorrelation coefficient for three prototype signals.

Figure 3.19 shows the autocorrelation coefficient at $\eta = 20$ for an example signal. The lower the input frequency and the more tonal the signal is, the higher the coefficient is. During the pauses, it drops toward 0.

Table 3.14 Autocorrelation coefficient for the three prototypical signal types *silence* (zero magnitude at all samples), *white noise*, and a sinusoidal signal

Input Signal	v_{ACF}
silence	not def.
white noise	≈ 0
sinusoidal	high

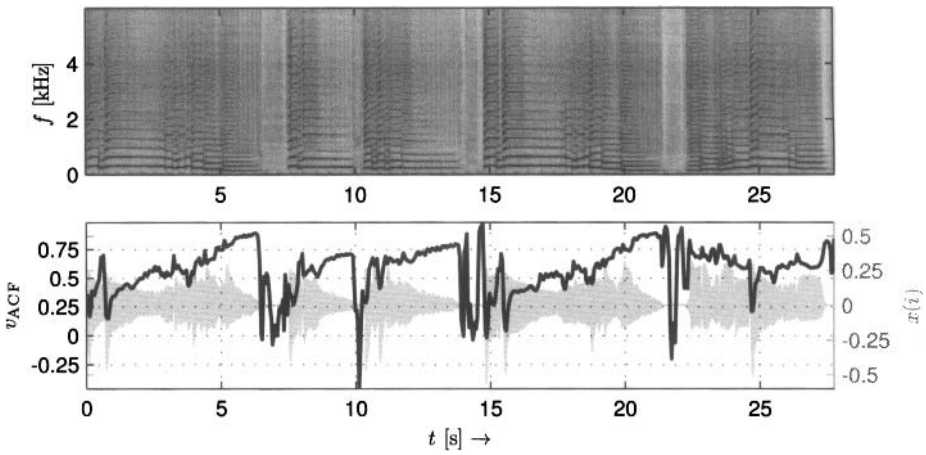

Figure 3.19 Spectrogram (top), waveform (bottom background), and autocorrelation coefficient 20 (bottom foreground) of a saxophone signal

3.4.3 Zero Crossing Rate

The number of changes of sign in consecutive blocks of audio samples — the *zero crossing rate* — is a low-level feature that has been used for decades in speech and audio analysis due to its simple calculation:

$$v_{\mathrm{ZC}}(n) = \frac{1}{2 \cdot K} \sum_{i=i_s(n)}^{i_e(n)} \left| \operatorname{sign}\left[x(i)\right] - \operatorname{sign}\left[x(i-1)\right] \right| \tag{3.58}$$

with the sign function being defined by

$$\operatorname{sign}\left[x(k)\right] = \begin{cases} 1, & \text{if } x(i) > 0 \\ 0, & \text{if } x(i) = 0 \\ -1, & \text{if } x(i) < 0 \end{cases} \tag{3.59}$$

and $x(i-1) = 0$ used as initialization if $x(i-1)$ does not exist.

The output will be a value in the range of $0 \leq v_{\mathrm{ZC}}(n) \leq 1$. The more often the signal changes its sign, the more high-frequency content can be assumed to be in the signal. Furthermore, the more the zero crossing rate varies over blocks, the less periodic the signal can be assumed to be. The concept of the zero crossing rate is based on input signals with an arithmetic mean of approximately 0.

Table 3.15 Zero crossing rate for the three prototypical signal types *silence* (zero magnitude at all samples), *white noise*, and a sinusoidal signal

Input Signal	v_{ZC}
silence	0
white noise	high
sinusoidal	period lengths per block times 2

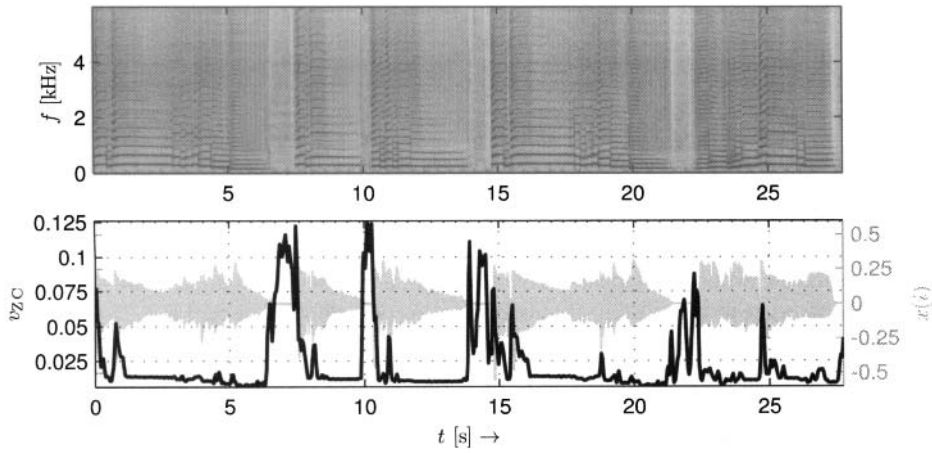

Figure 3.20 Spectrogram (top), waveform (bottom background), and zero crossing rate (bottom foreground) of a saxophone signal

Table 3.15 shows the results for the zero crossing rate for three prototype signals.

Figure 3.20 shows the zero crossing rate for an example signal. Unsurprisingly, it is high for noisy parts and low for tonal parts. Its usability for fundamental frequency detection (see Sect. 5.3.3.1) is indicated by the long constant values during constant pitches.

3.4.3.1 Common Variants

The zero crossing rate has been used for both measuring the tonalness of a signal as introduced in Sect. 3.4.1 and estimating its fundamental frequency by assuming a sinusoidal input signal and then relating the number of zero crossings directly to the fundamental frequency. To improve the robustness of such attempts, the input signal can be low-pass filtered to suppress high-frequency content interaction with the feature result. The cut-off frequency of the low-pass filter then should be chosen as low as the highest expected fundamental frequency to ensure maximum suppression of high-frequency content.

3.5 Feature Post-Processing

The result of the feature extraction process is a series of feature values that can — dependent on the use case — be processed, transformed, and selected.

It is quite common to compute a large number of features. Formally, they can be represented in a matrix:

$$\begin{aligned} \boldsymbol{V} &= [\boldsymbol{v}(0)\ \boldsymbol{v}(1)\ \ldots\ \boldsymbol{v}(\mathcal{N}-1)] \\ &= \begin{bmatrix} v_0(0) & v_0(1) & \ldots & v_0(\mathcal{N}-1) \\ v_1(0) & v_1(1) & \ldots & v_1(\mathcal{N}-1) \\ \vdots & \vdots & \ddots & \vdots \\ v_{\mathcal{F}-1}(0) & v_{\mathcal{F}-1}(1) & \ldots & v_{\mathcal{F}-1}(\mathcal{N}-1) \end{bmatrix} \end{aligned} \tag{3.60}$$

with the number of rows being the number of features $\mathcal{F}$ and the number of columns being the number of blocks $\mathcal{N}$. Each vector $\boldsymbol{v}(n)$ consists of $\mathcal{F}$ feature values at block n and is called an *observation*.

3.5.1 Derived Features

It is possible to generate new features from the previously extracted features. These new derived features do not have to replace the original features; they can be added as complementary features. Sometimes these derived features are called subfeatures, but we will use the term *subfeature* in a slightly different context as described in Sect. 3.5.3.

In some cases the detection of (sudden) changes of feature values is of special interest as it may mark the start or end of important segments, for example, note onsets and structural boundaries. A simple way of analyzing these changes is to compute the difference between consecutive feature results (which would be called *derivative* if it were a continuous function):

$$v_{j,\Delta}(n) = v_j(n) - v_j(n-1). \tag{3.61}$$

The resulting series $v_{j,\Delta}(n)$ is either one value shorter than the corresponding series $v_j(n)$ or an appropriate initialization for $v_j(-1)$ has to be defined. The time stamp $t_{\mathrm{s},\Delta}(n)$ of $v_{j,\Delta}(n)$ would be

$$t_{\mathrm{s},\Delta}(n) = \frac{t_{\mathrm{s}}(n) + t_{\mathrm{s}}(n-1)}{2} \tag{3.62}$$

with $t_s(n)$ being the time stamp of $v_j(n)$.

Computing the derivative has the character of a high-pass filter; it is also common to do the opposite, namely to smooth out $v_i(n)$ with a low-pass filter. This allows us to focus on the long-term variations of the feature. In general it is beneficial if the used filter has a zero phase or linear phase response in order to ensure correct timing properties, therefore either an MA filter as introduced in Sect. 2.2.1.1 can be used or any IIR filter applied twice forward and backward on the series to produce the low-pass filtered series $v_{j,\mathrm{LP}}(n)$ (see Sect. 2.2.1.2).

The features usually are the input of the second processing step of an ACA system, namely a classification system, a distance measure, or some other system for feature interpretation. Depending on the classifier and feature selection algorithm used and given a training database (compared to the number of features), it may be of interest to have as many raw input features as possible or at least to have a large feature data set from which to choose. While linear combinations of features are frequently already covered by the selection algorithm (see below), non-linear combinations such as the multiplication of two series of features can theoretically improve results in certain cases. An example would be

$$v_{jl}(n) = v_j(n) \cdot v_l(n). \tag{3.63}$$

The usage of such derived features is frequently met with only limited success as most of the information of interest can already be found in the original features.

3.5.2 Normalization and Mapping

When different features are combined into vectors $\boldsymbol{v}(n)$, their different output ranges and distributions might become a problem. This is, for example, the case when computing the Euclidean distance because one feature may have more impact on the result than another. Consider two identical features with identical distribution except for an amplitude scale factor λ. Each observation contains the two features. When computing the Euclidean distance, the second dimension's distance will have the weight λ^2 while the first dimension will be weighted with 1. Large λ will thus let the second feature dominate the distance while small λ will render the second dimension superfluous.

A common approach to normalize features if they all have *symmetric* distributions with *identical shape* is to remove their mean value and scale them to a variance of 1 (see, e.g., [66]):

$$v_{j,\mathrm{N}}(n) = \frac{v_j(n) - \mu_{v_j}}{\sigma_{v_j}}. \tag{3.64}$$

However, if the distributions do not have identical shape this normalization is not applicable. In that case, other approaches are necessary to ensure a correct combination of features.

3.5.2.1 Feature Distribution

To get similar feature distributions, a target distribution has to be chosen to which all the features will be transformed if necessary. In most cases, a Gaussian distribution is chosen as target.

Several approaches exist to transform a given distribution into a Gaussian distribution; widely used is the *Box-Cox transform* [67]. One example of this transform is

$$v^{(\lambda)} = \begin{cases} \frac{v^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(v), & \lambda = 0 \end{cases} \tag{3.65}$$

with the parameter λ to be estimated. The Box-Cox transform only considers a limited class of transformation functions and thus does not guarantee that any arbitrary distribution can be mapped to a Gaussian distribution.

There are also numerical methods of finding appropriate feature transformations. One example is the work of Albada and Robinson who transform arbitrary distributions to the normal distribution [68]. The transformation function for every feature has then to be stored numerically.

In many practical applications it is sufficient to transform selected features in a way that results only in roughly approximated Gaussian distributions. Only features failing a test for Gaussianity have to be subjected to a transformation. Statistical procedures to test a distribution for Gaussianity are, for example, the Kolmogorov-Smirnov test, the Lilliefors test, the Shapiro-Wilk test [69], and the Anderson-Darling test [70]. Thode gives a good introductory overview [71]. Unfortunately, the result of these statistical tests is only of limited use in ACA because these tests nearly always tend to fail for large numbers of observations. In these cases it is more practical to compute both the skewness (see Sect. 3.2.7)

and possibly the kurtosis (see Sect. 3.2.8) of the features to determine how *Gaussian* their distribution is. As a rule of thumb, distributions with a skewness smaller than 2 are not significantly skewed and can thus be assumed to be symmetric [72].

3.5.2.2 Feature Normalization

The standard approach to feature *normalization* has been given in Eq. (3.64). When the feature distribution is not shaped like a Gaussian distribution, it can make more sense to normalize it with respect to its median $Q_v(0.5)$ as described in Eq. (3.25). The normalized feature is then

$$v_{j,\mathrm{N}}(n) = \frac{v_j(n) - Q_{v_j}(0.5)}{s_{v_j}} \tag{3.66}$$

with s_{v_j} being the root mean squared deviation from the median

$$s_{v_j} = \sqrt{\frac{1}{\mathcal{N}} \sum_{n=0}^{\mathcal{N}-1} \left(v_j(n) - Q_{v_j}(0.5)\right)^2}. \tag{3.67}$$

The normalization of multi-dimensional features requires special consideration and depends on both the feature characteristics as well as the specific use case.

3.5.3 Subfeatures

Each feature still represents a time series, and it is common to compute a time-independent summary of each feature per sliding texture window (the term *texture window* has been introduced in Sect. 2.2.2). The summary feature is frequently called *subfeature*. The most common subfeatures would be the arithmetic mean and the standard deviation of the feature or, more general, basically every statistical measure presented in Sect. 2.1.4. Furthermore, it is basically possible to use each single feature presented above as a description of the feature time series by computing the feature of a feature, although certainly not every combination would make sense. A relatively large number of possible meaningful subfeatures in the context of musical genre classification was presented by Mörchen et al. [73].

3.5.4 Feature Dimensionality Reduction

Although large numbers of features can easily be extracted from the audio data, it is unclear a priori which features will help the final interpretation or classification stage of an ACA system. In other words, it is not known which features are of relevance.

The accuracy of a pattern recognition system or classifier will not necessarily improve with an increasing number of features. A large number of features increases the likelihood of *overfitting* to occur during the training. Overfitting means that the classifier starts to learn training set specific characteristics which cannot be generalized to the use case. The classifier will perform poorly on unknown input data. Therefore, the number of features has to be reduced and fitted to the amount of available training data.[3] The optimal number of features is difficult to determine; typically it is simply the number of features that maximize

[3]There exist classifiers which are relatively robust against such dimensionality issues while other classifiers are where susceptible. The danger of overfitting should be carefully considered for each combination of classifier type, the number of features, and the size of the training data set.

the classification accuracy of the training set using *N-fold cross validation* (see Sect. 8.1.3). To estimate the minimum feature performance, it is helpful to add an additional feature consisting of random noise to the feature set.

We can define the following criteria for a feature to be helpful:

- *high "discriminative" or descriptive power* since the feature should be suitable to the task at hand,
- *non-correlation to other features* because each feature should add new information to avoid redundancy,
- *invariance to irrelevancies* to allow the feature to be robust against, e.g., linear transformations of the input audio signal such as scaling and filtering operations (low-pass filtering, reverberation), the addition of signals such as (background) noise, coding artifacts as well as the application of non-linear operations such as distortion and clipping (see Wegener et al. for an example evaluation of feature robustness [74]), and
- *reasonable computational complexity* to ensure that the feature is able to be computed on the target platform (such as a mobile device) and for the required application, respectively.

We will mainly focus on the first two criteria; the third criterion can be easily tested by adding modified audio files to the test set, and the fourth criterion is too application-specific to be dealt with in a general way.

There are two different approaches to reduce the dimensionality of the feature space:

- *feature subset selection* to discard specific features, and
- *feature space transformation* to transform the features to a lower dimensional space.

In the first case the most promising feature subset is chosen; in the latter case only those dimensions in the transformed feature space are discarded that contribute the least information for the target application.

Note that the requirement of feature dimensionality reduction strongly depends on the classification algorithm used; furthermore, it is still under discussion whether complex methods of dimensionality reduction really outperform simpler ones [75].

3.5.4.1 Feature Subset Selection

The aim of *feature subset selection* is to reduce the number of used features by discarding the least powerful features. Formally, the available feature set

$$\mathcal{V} = v_j|_{j=1,\ldots,\mathcal{F}} \tag{3.68}$$

should be reduced to the feature subset

$$\mathcal{V}_\mathrm{s} = v_j|_{j=1,\ldots,\mathcal{F}_\mathrm{s}} \tag{3.69}$$

with $\mathcal{F}_\mathrm{s} < \mathcal{F}$; the subset is chosen to optimize a given objective function $J(\mathcal{V}_\mathrm{s})$.

Feature selection algorithms are called *wrapper methods* if the objective function is the classifier itself and *filter methods* if the objective function $J(\mathcal{V}_\mathrm{s})$ is independent of the classification system used. Filter methods select features based on properties a good feature set is presumed to have and are usually computationally less expensive than wrapper methods.

This section will only provide a short introduction to the most common approaches to feature subset selection. More in-depth surveys of this topic have been published by Guyon and Elisseeff [76] and Cantú-Paz et al. [77].

Examples for wrapper methods are

- *Brute Force Subset Selection*

  The most obvious way of finding the optimal feature subset is to compute the classification accuracy for all possible combinations of features and to select the subset that performed best. The disadvantage of this approach is that the number of subsets to test, i.e., to train and evaluate, will be $2^{\mathcal{F}}$. This renders this method impractical for large numbers of features $\mathcal{F}$.

- *Single Variable Classification*

  A simple feature ranking can be obtained by calculating the classification accuracy for each individual feature. This enables the identification of features performing very poorly individually. While discarding the features that perform worst seems to be an intuitive solution, there are two problems with selecting or discarding features this way:

  1. The feature ranking contains no information on the *correlation* of two or more features. Consider the case of two features with identical values. They will both have the same ranking which is possibly high, but leaving both in the selected feature subset cannot improve classifier performance (and might even harm it in the case of simple classification algorithms).
  2. The feature ranking contains no information on the *combined usefulness* of features. A feature that adds no information individually might be able to add information in combination with other features.

- *Sequential Forward Selection*

  Sequential forward selection starts with an empty subset of features. In the first iteration, it considers all feature subsets with only one feature similar to the *single variable classification* approach method. The subset with the highest classification accuracy is used as the basis for the next iteration. The iterative algorithm can be structured into the following processing steps:

  1. Start with an empty feature subset $\mathcal{V}_{\mathrm{s}} = \emptyset$.
  2. Find the one feature v_j not yet included in the feature subset that maximizes the objective function
  $$v_j = \underset{\forall j | v_j \notin \mathcal{V}_{\mathrm{s}}}{\operatorname{argmax}} J(\mathcal{V}_{\mathrm{s}} \bigcup v_j). \tag{3.70}$$
  3. Add feature v_j to $\mathcal{V}_{\mathrm{s}}$.
  4. Go to step 2 and repeat the procedure until the required number of features has been selected or the required classification accuracy has been reached.

- *Sequential Backward Elimination*

  Sequential backward elimination works in an analogous way to sequential forward selection but starts with a full subset of features and iteratively removes features from the subset. It is computationally less efficient than sequential forward selection. This is particularly true for large feature sets. Sequential backward elimination can be argued to give better results since sequential forward selection does not assess the importance of features in combination with other not yet included features.

There exist many filter methods for finding variable rankings. The methods include, for example, chi-square statistics or any arbitrary class separability measure. One common example of a filter yielding an implicit feature ranking is *Principal Component Analysis (PCA)*. The concepts of PCA are summarized in Appendix C. It can be used for feature selection by examination of the transformation matrix $\boldsymbol{T}$. The idea is to keep the features with major influence on the principal components and to eliminate features with major influence on the components with low variance.

A simple rule for feature elimination is to start with the component with the smallest eigenvector, discard the feature which contributes most to this component, proceed to the next-smallest eigenvector, and repeat the procedure until all features have been ranked. Then, an arbitrary number of features can be discarded.

3.5.4.2 Feature Space Transformation

The objective of *feature space transformation* is to reduce the number of used features by transforming them into a lower dimensional space.

The disadvantage of using transformations for dimensionality reduction is that the transformed features cannot be interpreted as easily as the original features since the transformed features are linear combinations of the original features. Examples of tools for feature space transformation are

- *Principal Component Analysis*

  Transforming the data set with PCA (see Appendix C) does not reduce the dimensionality of the data by itself. However, the new dimensions can be sorted according to the variance they contribute to the data which can be seen as a measure of importance. Discarding the components that account for low variance is therefore a viable approach.

  A widely used systematic criterion to decide how many components can be discarded is based on the eigenvalue. This approach is based on the assumption that every component with an eigenvalue lower than 1 can be discarded. This criterion is equivalent of a threshold of $1/\mathcal{F}$ for the relative variance for which a component accounts.

  In many cases, this criterion leaves more components in the data set than useful. A slightly more "hands-down" approach is to identify either the index after which the eigenvalues are significantly lower or the index after which eigenvalues tend to be very similar to each other.

- *Other Transformation Methods*

  Other transformation methods can be used for feature space transformation but will not be explained in detail in this book. Typical approaches that can be found in the literature are *Independent Component Analysis (ICA)* and *Singular Value Decomposition (SVD)*. *Linear Discriminant Analysis (LDA)* also transforms the feature space, however, the number of output components cannot be chosen freely in this case. The main distinction of LDA and PCA is that PCA maximizes the variance and LDA maximizes class separability.