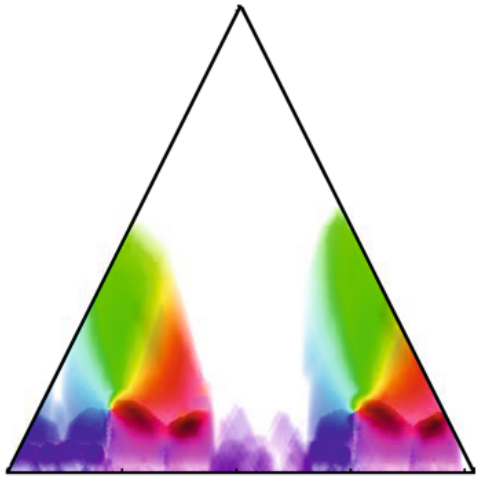
Chapter 4
Music Structure Analysis



One of the attributes distinguishing music from random sound sources is the hierarchical structure in which music is organized. At the lowest level, one has events such as individual notes, which are characterized by the way they sound, their timbre, pitch, and duration. Combining various sound events, one obtains larger structures such as motifs, phrases, and sections, and these structures again form larger constructs that determine the overall layout of the composition. This higher structural level is also referred to as the musical structure of the piece, which is specified in terms of musical parts and their mutual relations. For example, in popular music such parts can be the intro, the chorus, and the verse sections of the song. Or in classical music, they can be the exposition, the development, and the recapitulation of a movement. The general goal of **music structure analysis** is to divide a given music representation into temporal segments that correspond to musical parts and to group these segments into musically meaningful categories.

Let us consider a concrete example. Figure 4.1a shows a sheet music representation of the Mazurka Op. 6, No. 4 by the Polish composer Frédéric Chopin. This piano piece can be subdivided into five sections, where the third and fifth sections are repetitions of the first section. Therefore, these sections belong to the same category denoted by the symbol A. Similarly, the fourth section is a repetition of the second one. These two sections belong to another group labeled by the symbol B. Hence, at an abstract level, the overall musical structure can be described by the sequence $A_1B_1A_2B_2A_3$ (see Figure 4.1d). Instead of using the musical score, one typical scenario is to derive structural information from a given audio recording

M. Müller, *Fundamentals of Music Processing*,
DOI 10.1007/978-3-319-21945-5_4

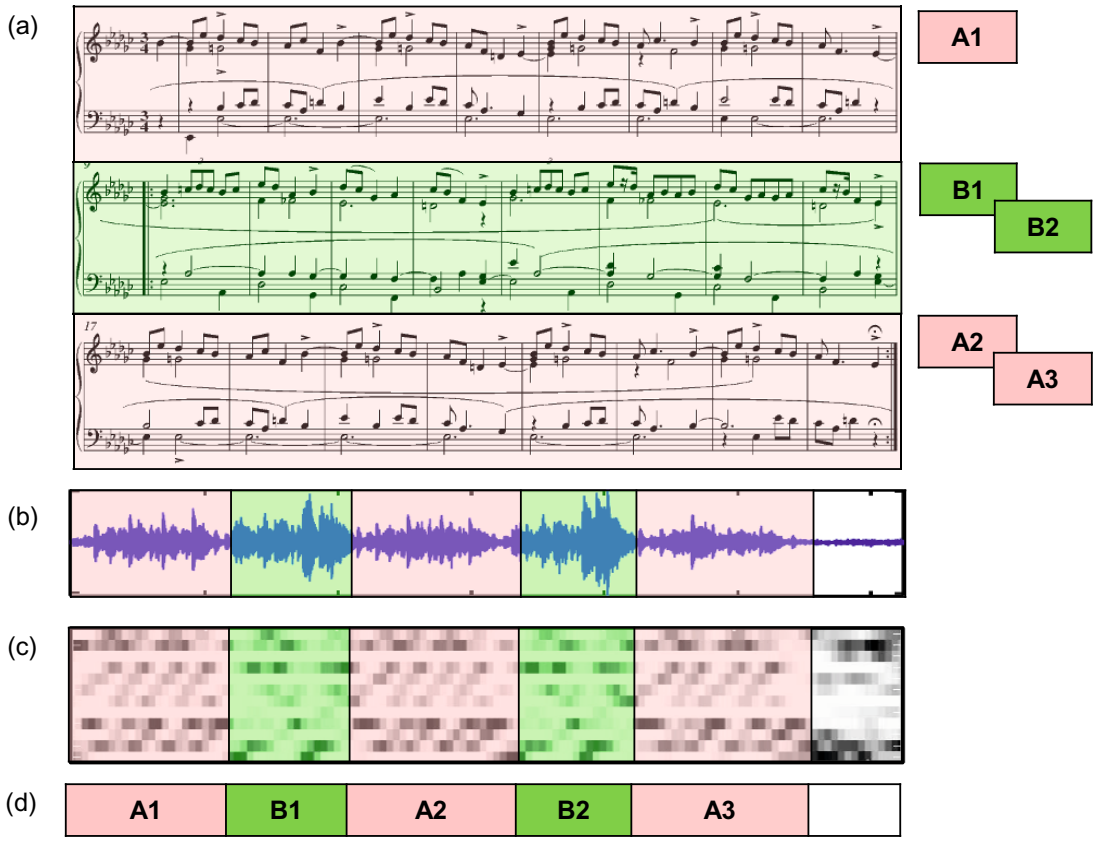


Fig. 4.1 Musical structure of the Mazurka Op. 6, No. 4 by Chopin. **(a)** Sheet music representation. **(b)** Waveform of an audio recording. **(c)** Chroma representation derived from (b). **(d)** Manually annotated segmentation of the audio recording.

(see Figure 4.1b). To this end, one needs to convert the waveform into a suitable feature representation that captures musical properties relevant for the structure of interest. In our example, as shown by Figure 4.1c, the repetition-based structure can be seen in a chroma representation that captures harmonic information.

As demonstrated by the previous example, the musical structure is often related to recurring patterns such as repeating sections. In general, however, there are many more criteria for segmenting and structuring music. For example, certain musical sections may be characterized by some homogeneity property such as a consistent timbre, the presence of a specific instrument, or the usage of certain harmonies. Furthermore, segment boundaries may go along with sudden changes in musical properties such as tempo, dynamics, or the musical key. These various segmentation principles require different methods, which may be loosely categorized into repetition-based, homogeneity-based, and novelty-based approaches.

In this chapter, we study general techniques for deriving structural information from a given music recording. In Section 4.1, we start by giving an overview of different segmentation principles, while introducing a working definition of the structure analysis problem as used in the subsequent sections. Furthermore, we discuss some feature representations that account for different musical dimensions. The con-

cept of self-similarity matrices, which we study in Section 4.2, is of fundamental importance in computational music structure. In particular, we show how the various segmentation principles are reflected in such matrices and how this can be exploited for deriving structural information. As a first application of self-similarity matrices, we discuss in Section 4.3 a subproblem of music structure analysis known as audio thumbnailing. The goal of this problem is to determine the audio segment that best represents a given music recording. Providing a compact preview, such audio segments are useful for music navigation applications similar to visual thumbnails that help in organizing and accessing large photo collections. While we apply repetition-based principles for audio thumbnailing, we discuss in Section 4.4 some segmentation procedures that rely on novelty-based principles. The objective of such procedures is to specify points within a given audio recording where a human listener would recognize a change, a sudden event, or the transition between two contrasting parts. Finally, in Section 4.5, we address the issue of evaluating analysis results, which itself constitutes a nontrivial problem.

4.1 General Principles

Music structure analysis is a multifaceted and often ill-defined problem that depends on many different aspects. First of all, the complexity of the problem depends on the kind of music representation to be analyzed. For example, while it is comparatively easy to detect certain structures such as repeating melodies in sheet music, it is often much harder to automatically identify such structures in audio representations. Second, there are various principles including homogeneity, repetition, and novelty that a segmentation may be based on. While the musical structure of the piano piece shown in Figure 4.1 is based on repetition, musical parts in other music may be characterized by a certain instrumentation or tempo. Third, one also has to account for different musical dimensions, such as melody, harmony, rhythm, or timbre. For example, in Beethoven's Fifth Symphony the "fate motif" is repeated in various ways—sometimes the motif is shifted in pitch; sometimes only the rhythmic pattern is preserved. Finally, the segmentation and structure largely depend on the musical context and the temporal hierarchy to be considered. For example, the recapitulation of a sonata may be considered a kind of repetition of the exposition on a coarse temporal level even though there may be significant modifications in melody and harmony on a finer temporal level. Figure 4.2 gives an overview of various aspects that need to be considered when dealing with musical structures. In the following, we discuss these aspects in more detail. In particular, our goal is to raise the awareness that computational procedures as described in the subsequent sections are often based on simplifying model assumptions that only reflect certain aspects of the complex structural properties of music.

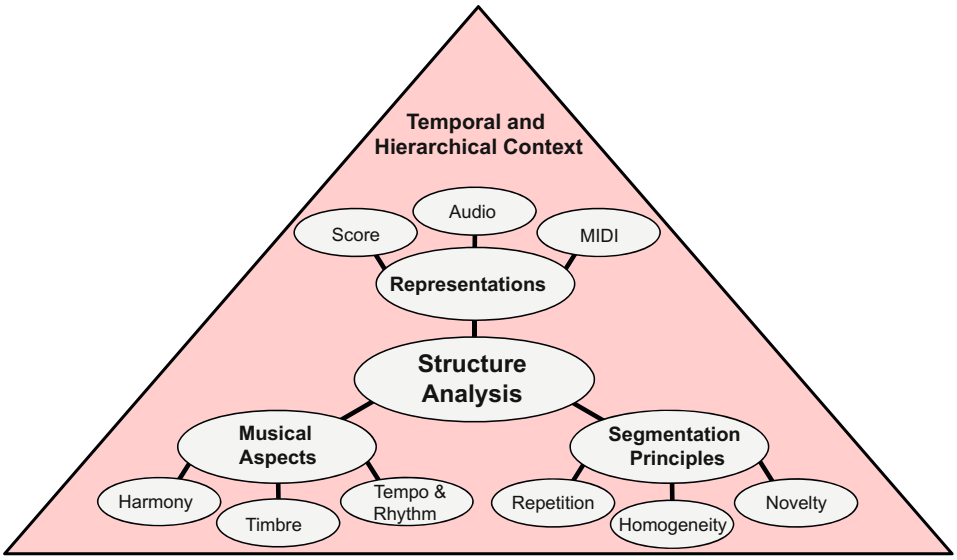


Fig. 4.2 Overview of various segmentation and structure principles.

4.1.1 Segmentation and Structure Analysis

The tasks of segmenting and structuring multimedia documents are of fundamental importance not only for the processing of music signals but also for general audio-visual content. **Segmentation** typically refers to the process of partitioning a given document into multiple segments with the goal of simplifying the representation into something that is more meaningful and easier to analyze than the original document. For example, in image processing the goal is to partition a given image into a set of regions such that each region is similar with respect to some characteristic such as color, intensity, or texture (see Figure 4.3 for an illustration). Region boundaries can often be described by contour lines or edges at which the image brightness or other properties change sharply and reveal discontinuities. In music, the segmentation task is to decompose a given audio stream into acoustically meaningful sections each corresponding to a continuous time interval that is specified by a start and end boundary. At a fine level, the segmentation may aim to find the boundaries between individual notes or to find the beat intervals specified by beat positions. At a coarser level, the goal may be to detect changes in instrumentation or harmony or to find the boundaries between verse and chorus sections. Also, discriminating between silence, speech, and music, finding the actual beginning of a music recording, or separating the applause at the end of a performance are typical segmentation tasks.

Going beyond mere segmentation, the goal of **structure analysis** is to also find and understand the relationships between the segments. For example, certain segments may be characterized by the instrumentation. There may be sections played only by strings. Sections played by the full orchestra may be followed by solo sections. The verse sections with a singing voice may be alternated with purely instrumental sections. Or a soft and slow introductory section may precede the main theme played in a much faster tempo. Furthermore, sections are often repeated. Most

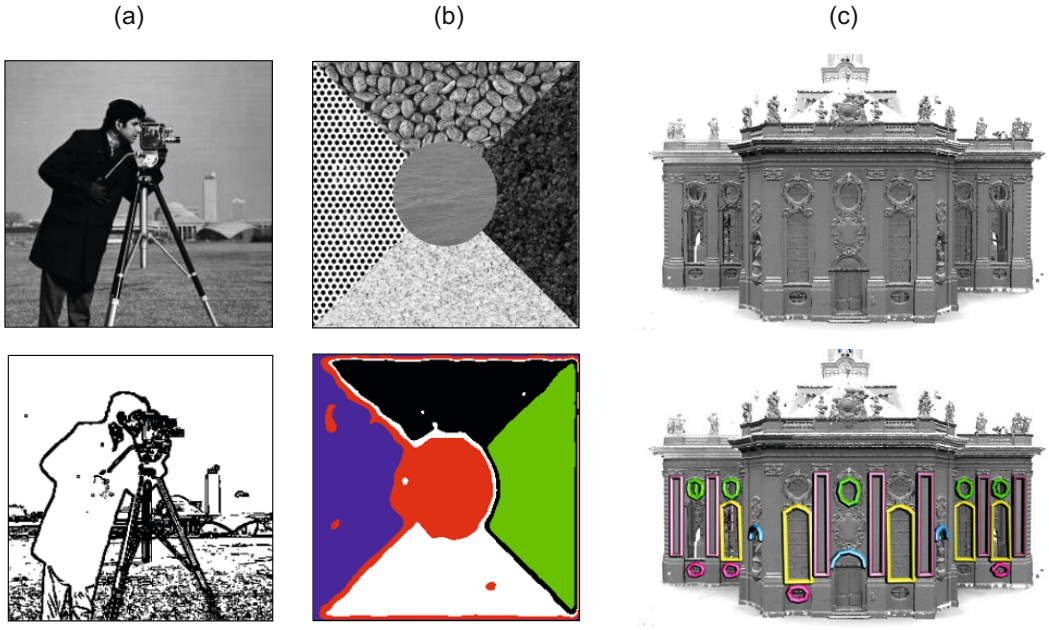


Fig. 4.3 Examples for segmentation results for image and 3D data. **(a)** Novelty-based image segmentation using edge detection. **(b)** Homegeneity-based texture segmentation. **(c)** Repetition-based segmentation of 3D geometry (from [66]).

events of musical relevance are repeated in a musical work in one way or another. However, repetitions are rarely identical copies of the original section, but undergo modifications in aspects such as the lyrics, the instrumentation, or the melody. One main task of structure analysis is to not only segment the given music recording, but to also group the segments into musically meaningful categories (e.g., intro, chorus, verse, outro).

The challenge in computational music structure analysis is that structure in music arises from many different kinds of relationships including repetition, contrast, variation, and homogeneity [53]. As we have already noted, **repetitions** play a particularly important role in music, where sounds or sequences of notes are often repeated [39]. Recurrent patterns can be of rhythmic, harmonic, or melodic nature. On the other hand, **contrast** is the difference between successive musical sections of different character. For example, a quiet passage may be contrasted by a loud one, a slow section by a rapid one, or an orchestral part by a solo. A further principle is that of **variation**, where motifs and parts are picked up again in a modified or transformed form. Finally, a section is often characterized by some sort of inherent **homogeneity**; for example, the instrumentation, the tempo, or the harmonic material may be similar within the section. All these principles need to be considered in the temporal context. Music happens in time (as opposed to, say, a painting), and it is the **temporal order** of events that is essential for building up musically and perceptually meaningful entities such as melodies or harmonic progressions [3].

In view of the various principles that crucially influence the musical structure, a large number of different approaches to music structure analysis have been developed. In this chapter, we want to roughly distinguish three different classes of

methods. First, **repetition-based** methods are used to identify recurring patterns. Second, **novelty-based** methods are employed to detect transitions between contrasting parts. Third, **homogeneity-based** methods are used to determine passages that are consistent with respect to some musical property. Note that novelty-based and homogeneity-based approaches are two sides of a coin: novelty detection is based on observing some surprising event or change after a more homogeneous segment. While the aim of novelty detection is to locate the changes' time positions, the focus of homogeneity analysis lies in the identification of longer passages that are coherent with respect to some musical property. In the following section, we will study various procedures for structure analysis following one or several of these paradigms.

4.1.2 Musical Structure

As already mentioned in the introduction of this chapter, our focus is to analyze a given music recording on a rather coarse structural level. This level corresponds to what is often referred to as the **musical structure**, which describes the overall structural layout of a piece of music. In particular for Western classical music, one also encounters the term **musical form**, which refers to specific structural categories exploiting the principles of contrast and variety in one way or another. In this chapter, we use the term "musical structure" loosely, including with it the concept of musical form.

To specify musical structures, we now introduce some terminology as used in the remainder of this book. First of all, we want to distinguish between a piece of music (in an abstract sense) and a particular audio recording (an actual performance) of the piece. The term **part** is used in the context of the abstract music domain, whereas the term **segment** is used for the audio domain. Furthermore, we use the term **section** in a rather vague way for both domains to denote either a segment or a part. Musical parts are typically denoted by the capital letters $A, B, C, \ldots$ in the order of their first occurrence, where numbers (often written as subscripts) indicate the order of repeated occurrences. For example, the sequence $A_1 B_1 A_2 B_2 A_3$ describes the musical structure of the piano piece shown in Figure 4.1, which consists of three repeating A-parts and two repeating B-parts. Hence, given a recording of this piece of music, the goal of the structure analysis problem (as considered in this chapter) is to find the segments within the recording that correspond to the A- and B-parts.

In Western music, the musical structure often follows certain structural patterns (see Figure 4.4). The simplest of these patterns is the **strophic form**, which basically consists of a sequence of a part being repeated over and over again. The form $A_1 A_2 A_3 A_4 \ldots$ is, for example, used in folk songs or nursery rhymes, where the A-parts correspond to the stanzas of the underlying poem. Another structural pattern is referred to as **chain form**, which is simply a sequence of self-contained and unrelated parts ($ABCD\ldots$), sometimes with repeats ($A_1 A_2 B_1 B_2 C_1 C_2 D_1 D_2 \ldots$). This form is often used in a composition that consists of a concatenation of favorite tunes from

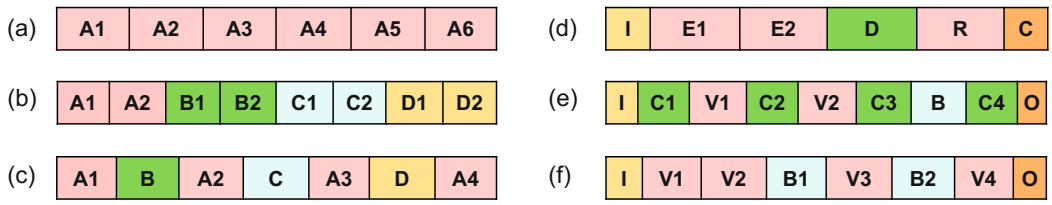


Fig. 4.4 Examples for musical structures as encountered in Western music. **(a)** Strophic form. **(b)** Chain form with repetitions. **(c)** Rondo form. **(d)** Sonata form. **(e)** Beatles song "Tell Me Why." **(f)** Beatles song "Yesterday."

popular songs, dances, or operettas. Examples are **medleys** or **potpourris**, which are pieces composed from parts of existing pieces that are simply juxtaposed with no strong connection or relationship. Another form is the **rondo form**, where a recurring theme alternates with contrasting sections, yielding the musical structure $A_1BA_2CA_3DA_4\ldots$.

In Western classical music, one of the most important musical structures is known as the **sonata form**, which is a large-scale musical structure typically used in the first movements of sonatas and symphonies. The basic sonata form consists of an **exposition** (E), a **development** (D), and a **recapitulation** (R), where the exposition is repeated once. Sometimes, one can find an additional **introduction** (I) and a closing **coda** (C), thus yielding the form IE_1E_2DRC. In particular, the exposition and the recapitulation stand in close relation to each other, both containing two subsequent contrasting subject groups (often simply referred to as the first and second theme) connected by some transition. As previously noted, at least at a coarse level, the recapitulation can be regarded as a kind of repetition of the exposition. However, at a finer level, there are significant differences. For example, the subject groups and transition in the recapitulation are musically altered and can be quite different from their corresponding occurrences in the exposition. Finally, we want to discuss some typical structural elements one finds in popular music. As with the sonata form, one sometimes uses generic names to denote the musical parts instead of using capital letters. The most important parts of a pop song are the **verse** (V) and the **chorus** (C) sections. Each verse usually employs the same melody (possibly with slight modifications), while the lyrics change for each verse. The chorus (sometimes also called the **refrain**) typically consists of a melodic and lyrical phrase which is repeated. Sometimes, pop songs may start with an **intro** (I) and close with an **outro** (O). Finally, verse and chorus sections may be connected by an additional part called a **bridge** (B). The verse and chorus are usually repeated throughout a song, while the intro and the outro appear only once. Some pop songs may have a **solo** section, where one or more instruments play a melodic line, typically following the melody previously introduced by the singer.

We have presented only a small selection of musical structures. In practice, there are many more structures as well as variations and deviations from standard forms as illustrated by the last two examples of Figure 4.4. A musical structure can be rather vague, and even music experts may argue about the construction of a given compo-

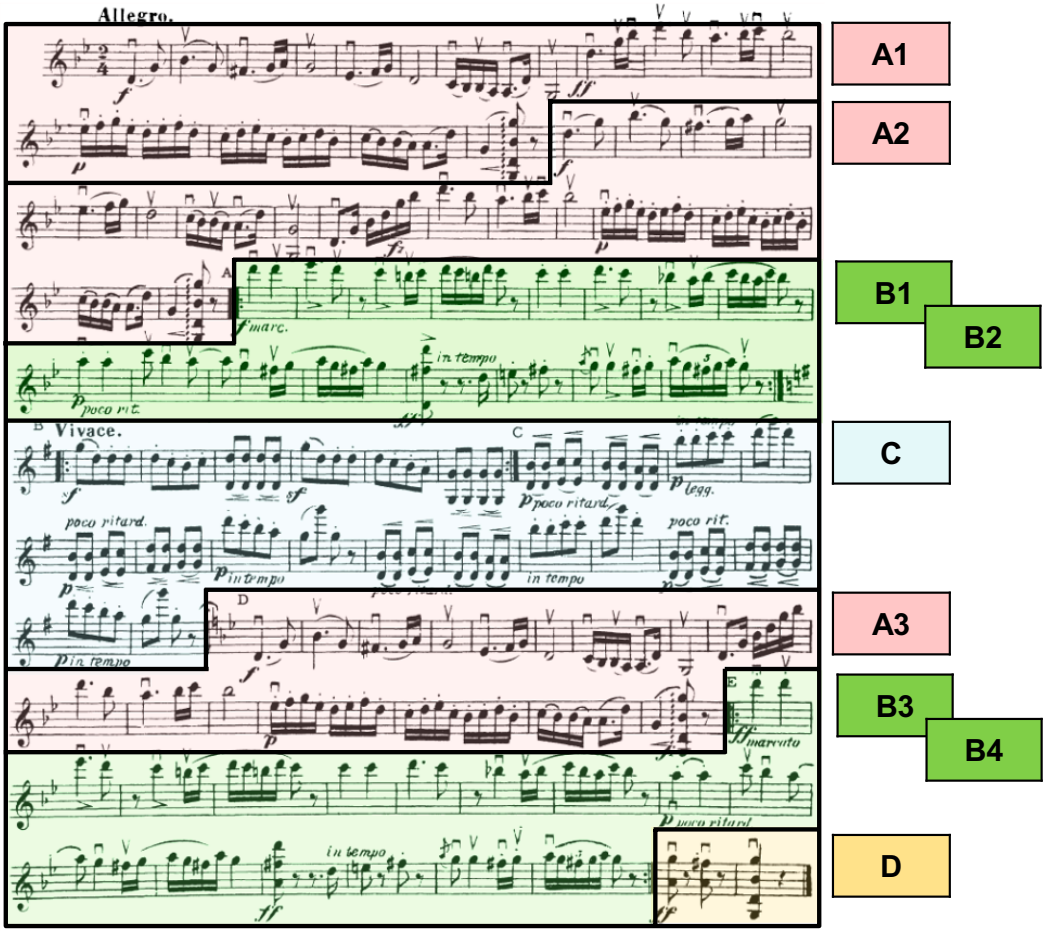


Fig. 4.5 Sheet music representation and musical structure of the Hungarian Dance No. 5 by Johannes Brahms. Only the voice for the violin of an arrangement for full orchestra is shown.

sition. In particular, what we call a repetition of a musical section is often far from being an exact copy. Segments that are considered to correspond to the same musical part may differ in instrumentation and tempo, or a segment may be transposed to another key, the melody may be changed while only the underlying harmonic progression is kept, and so on. Furthermore, musical structure is typically ordered in hierarchies, and it is often not clear which level should be considered when specifying the musical structure. For example, in the piece shown in Figure 4.1, the *A*-part can be further subdivided into substructures consisting of two or even four subparts. Similarly, the *B*-part can be regarded as a repetition of two subparts. These repeating substructures also become visible in the chroma representation derived from the music recording (see Figure 4.1c). In music notation, such subparts are often indicated using small letters $a, b, c, \ldots$.

As a final example, we want to consider the Hungarian Dance No. 5 by Johannes Brahms, which will also serve as our running example in the next sections. This piece is part of a set of 21 dance tunes composed by Brahms up to 1869 and based mostly on traditional Hungarian themes. Each dance has been arranged for a wide variety of instruments and ensembles, ranging from piano versions to versions for

full orchestra. Figure 4.5 shows a sheet music representation for the violin voice of an arrangement for full orchestra. The musical structure as indicated in the figure is $A_1A_2B_1B_2CA_3B_3B_4D$, which consists of three repeating A-parts, four repeating B-parts, as well as a C-part and a short closing D-part. The A-part has a substructure consisting of two more or less repeating subparts. Furthermore, as becomes apparent when looking at the musical score, the middle C-part may be further subdivided into a substructure that may be described by $d_1d_2e_1e_2e_3e_4$ (see Figure 4.28).

The overall musical structure of this piece can be explained in terms of repeating elements. However, there are also many other musical cues that reinforce the musical structure. For example, the C-part stands in contrast to the remaining parts. First, there is a change of the musical key in the C-part (changing from G minor to G major). Then, there is a change in the notated tempo (changing from 'Allegro' to 'Vivace'). While the A- and B-parts have catchy tunes, there is no such melody in the C-part. Instead, the entire C-part is rather homogeneous with regard to harmony. However, this does not hold for other musical properties such as dynamics and tempo. For example, while the d-part segments are played in forte, the e-part segments are played in piano. Also there are many sudden tempo changes within the C-part. Therefore, in this case, a novelty-based segmentation procedure using tempo cues may be used to reveal the substructures of the C-part, whereas a homogeneity-based segmentation procedure using harmonic properties may be suited to distinguish the C-part from the other parts. We further develop this example in the next sections.

4.1.3 Musical Dimensions

We have already seen that the applicability of the different segmentation principles very much depends on the musical and acoustic properties of the audio signal to be analyzed. Since the sampled waveform of an audio signal is relatively uninformative by itself, the first step in automated structure analysis is to transform the given music recording into a suitable feature representation. As explained in the music synchronization scenario (Section 3.1), finding such a representation constitutes a delicate trade-off between robustness and expressiveness. Also, it is often unclear which musical properties are actually relevant for the given music signal and the considered segmentation scenario. For example, structural boundaries may be based on changes in harmony, timbre, or tempo. One major task in music processing is to transform a given audio signal into feature representations that correlate to the various musical aspects. In the following, we discuss this issue in more detail by considering three conceptually different feature representations (see Figure 4.6 for an overview).

As a first representation, we consider chroma features as introduced in Section 3.1.2. Recall that a normalized chroma vector describes the signal's local energy distribution over an analysis window (frame) across the twelve pitch classes of the equal-tempered scale (ignoring octave information). Capturing pitched content, a chroma-based feature sequence relates to harmonic and melodic properties

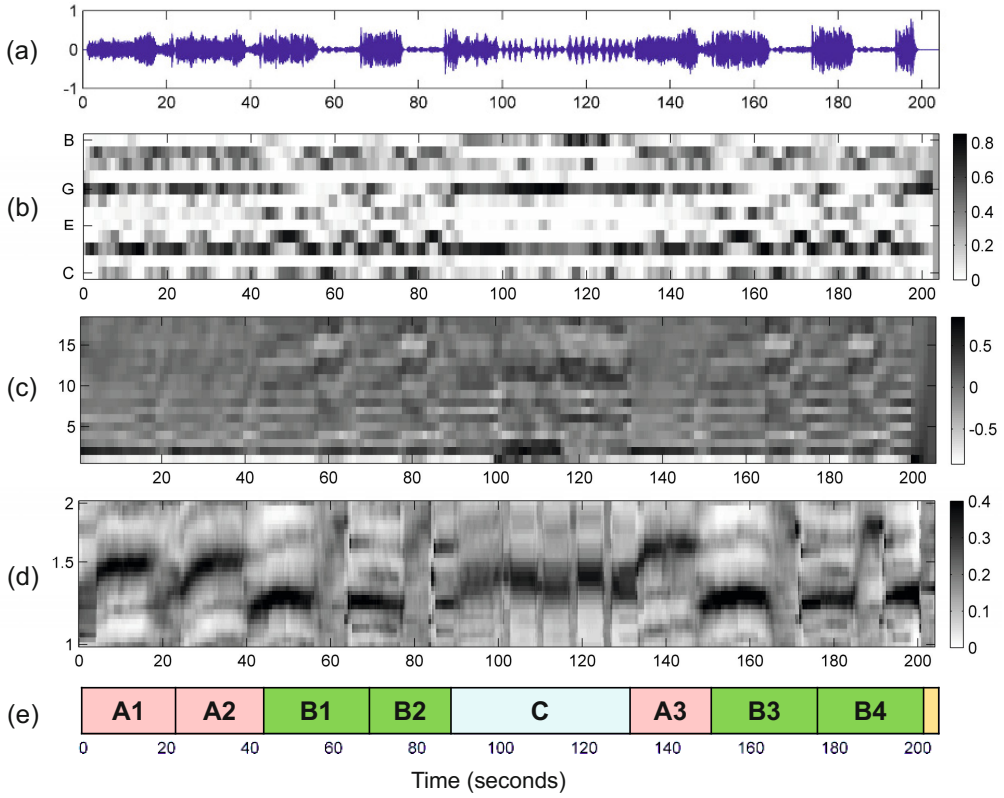


Fig. 4.6 Feature representations for a recording of the Hungarian Dance No. 5 by Johannes Brahms. **(a)** Waveform. **(b)** Chroma-based features. **(c)** MFCC-based features. **(d)** Tempo-based features. **(e)** Manually generated annotation.

of the music recording. Figure 4.6b shows a chroma representation derived from a recorded performance of our Brahms example, the Hungarian Dance No. 5. The patterns visible in the chromagram reveal important structural information. For example, the four repeating *B*-part segments are clearly visible as four similar characteristic subsequences in the chromagram. Furthermore, the *C*-part segment stands out in the chromagram by showing a high degree of homogeneity throughout the entire section. Indeed, for all chroma features of this segment, most of the signal's energy is contained in the G-, B-, and D-bands (which is not surprising since the *C*-part is in G major). In contrast, as for the *A*-part segments, many chroma vectors have dominant entries in the G-, B$^\flat$-, and D-bands (which nicely reflects that this part is in G minor).

Besides melody and harmony, the instrumentation and timbral characteristics are of great importance for the human perception of music structure. As we have discussed in Section 1.3.4, timbre is a rather vaguely defined perceptual property of sound, which is hard to describe and to extract from a music recording. For example, the automated recognition of musical instruments within polyphonic music signals is an extremely difficult problem. In applications such as structure analysis, it is often unnecessary to determine such information explicitly. Instead, mid-level representations that somehow correlate to aspects such as instrumentation and timbre

may be sufficient. In the context of timbre-based structure analysis, one often uses **mel-frequency cepstral coefficients** (MFCCs), which were originally developed for automated speech recognition. Parametrizing the rough shape of the spectral envelope, MFCC-based features capture timbral properties of the signal. At this point, we do not want to give a technical description on how these features are computed. Instead, let us have a look at Figure 4.6c, which shows an MFCC-based feature representation for our Brahms example. One can recognize that MFCC features within the *A*-part segments are different from the ones in the *B*-part and *C*-part segments. For many music recordings such as pop songs, where sections with singing voice alternate with purely instrumental or percussive sections, MFCC-based feature representations are well suited for novelty-based and homogeneity-based segmentation.

As a third musical dimension, we consider properties that are related to beat, tempo, and rhythmic information. Estimation of the tempo and beat positions is one of the central topics in music processing, which we cover in Chapter 6. In the music segmentation context, such techniques are often applied to derive **beat-synchronous** feature representations, where the time axis is segmented according to musically meaningful beat positions. Such beat-synchronous representations are very useful to compensate for tempo changes in repeating parts. On the downside, beat tracking errors introduced by automated procedures may have negative consequences for the subsequent music processing tasks to be solved (see Section 6.3.3 for more details).

In music structure analysis, tempo and beat information may also be used in combination with homogeneity-based segmentation approaches. Instead of extracting such information explicitly, a mid-level feature representation that correlates to tempo and rhythm may suffice for deriving a meaningful segmentation at a higher structural level. As an example, Figure 4.6d shows such a mid-level representation, a **tempogram**, which encodes local tempo information. More precisely, a cyclic variant of a tempogram is shown, where tempi differing by a power of two are identified—similar to cyclic chroma features, where pitches differing by octaves are identified. Technical details on how to compute such tempograms can be found in Section 6.2.4. Having a look at Figure 4.6d, one can notice that the different musical parts are played in different tempi (even though the representation does not reveal the exact tempi). Furthermore, there are sections where the tempogram features do not have any dominating entries, which may indicate that there is no clear notion of a tempo in the recording. This kind of information is also important and can be used for segmentation purposes. As this example indicates, a tempogram may yield information that is complementary to the information obtained by chroma-based or MFCC-based feature representations.

Besides the various musical dimensions, there is another aspect one should keep in mind when looking for suitable feature representations: the temporal dimension. In all of the above-mentioned feature representations, an analysis window is shifted over the music signal. As we have already seen for the STFT in Section 2.5.2, the length of the analysis window as well as the hop size parameter have a crucial influence on the quality of the feature representation. For example, long window sizes and large hop sizes may be beneficial for smoothing out irrelevant local variations,

which is often a desired property in homogeneity-based segmentation. On the downside, the temporal resolution decreases and important details may get lost, which can lead to problems when locating the exact segmentation boundaries.

In summary, a suitable choice of feature representations and parameter settings very much depends on the application context. Humans constantly and often unconsciously adapt themselves to the musical and acoustic characteristics of what they listen to. The richness and variety of musical structures make computational structure analysis a challenging problem.

4.2 Self-Similarity Matrices

We have seen that the principles of repetition, homogeneity, and novelty are fundamental for partitioning a given audio recording into musically meaningful structural elements. To study musical structures and their mutual relations, one general idea is to convert the music signal into a suitable feature sequence and then to compare each element of the feature sequence with all other elements of the sequence. This results in a **self-similarity matrix** (SSM), a tool which is of fundamental importance not only for music structure analysis but also for the analysis of many kinds of time series. In this section, we look at these matrices in detail. As we will see, one crucial property of self-similarity matrices is that repetitions typically yield path-like structures, whereas homogeneous regions yield block-like structures. These structural elements are exploited by most algorithms for visualizing, analyzing, and computing musical structures in one way or another. In Section 4.2.1, we introduce the concept of self-similarity matrices and discuss their basic structural properties. For applications, the improvement of these properties at an early state of the processing pipeline is of great importance, which is the topic of Section 4.2.2.

4.2.1 Basic Definitions and Properties

As said before, the concept of self-similarity matrices is fundamental for capturing structural properties of music recordings. Generally, one starts with a feature space $\mathcal{F}$ containing the elements of the feature sequence under consideration as well as with a similarity measure

$$s : \mathcal{F} \times \mathcal{F} \to \mathbb{R} \tag{4.1}$$

that makes it possible to compare these elements. Typically, the value $s(x,y)$ is high in case the elements $x,y \in \mathcal{F}$ are similar and small otherwise. Given a feature sequence $X = (x_1, x_2, \ldots, x_N)$, the idea is to compare all elements of the sequence with each other. This results in an N-square **self-similarity matrix** $\mathbf{S} \in \mathbb{R}^{N \times N}$ defined by

$$\mathbf{S}(n,m) := s(x_n, x_m), \tag{4.2}$$

where $x_n, x_m \in \mathcal{F}$, $n, m \in [1:N]$. In the following, a tuple $(n,m) \in [1:N] \times [1:N]$ is also called a **cell** of $\mathbf{S}$, and the value $\mathbf{S}(n,m)$ is referred to as the **score** of the cell (n,m).

Obviously, the concept of self-similarity matrices is closely related to the concept of cost matrices, which we have already encountered in Section 3.2.1. However, instead of a cost measure c as in (3.12), we now use a similarity measure s. And instead of comparing two sequences X and Y with each other, we now compare a single sequence X with itself. Depending on the application context and notion that is used to compare the data, there are many related concepts known under different names such as recurrence plot or self-distance matrix just to name a few. In this chapter, we only consider self-similarity matrices, but the techniques to be explained can easily be transferred to other types of matrices.

In the following discussion, we assume that the feature space is a Euclidean space $\mathcal{F} = \mathbb{R}^D$ of some dimension $D \in \mathbb{N}$. For simplicity and illustration purposes, we use as similarity measure s the absolute value of the inner product defined by

$$s(x,y) := |\langle x|y\rangle| \tag{4.3}$$

for two vectors $x, y \in \mathcal{F}$ (see (2.37)). With this similarity measure, the score between two orthogonal feature vectors is zero and otherwise it is positive. In the case that the feature vectors are normalized with respect to the Euclidean norm, the similarity values $s(x,y)$ lie in the interval $[0,1]$. Obviously, there are many more possibilities to define a similarity measure (see Exercise 4.1). The suitability of a similarity measure depends on the properties of the considered features and vice versa.

Given a feature sequence $X = (x_1, x_2, \ldots, x_N)$, it seems reasonable to require that an element x_n should be maximally similar to itself. Using normalized features and the similarity measure from (4.3), the similarity measure assumes its maximal value $s(x_n, x_n) = 1$ for all $n \in \mathbb{N}$. Therefore, the resulting SSM has a diagonal with large values. More generally, recurring patterns of the given feature sequence become visible in the SSM in the form of structures with large similarity values. The two most prominent structures induced by such patterns are often referred to as blocks and paths (see Figure 4.7a for an illustration). First, if the feature sequence captures musical properties that stay somewhat constant over the duration of an entire musical part, each of the feature vectors is similar to all other feature vectors within this segment. As a result, an entire **block** of large values appears in the SSM. In other words, homogeneity properties correspond to block-like structures. Second, if the feature sequence contains two repeating subsequences (e.g., two segments corresponding to the same musical part), the corresponding elements of the two subsequences are similar to each other. As a result, a **path** (or **stripe**) of high similarity running parallel to the main diagonal becomes visible in the SSM. In other words, repetitive properties correspond to path-like structures.

Before we further formalize these properties, let us have a look at Figure 4.7, which shows different self-similarity matrices for our Brahms example. Figure 4.7a shows an idealized SSM. For example, assuming that the three repeating A-part segments are homogeneous, the SSM has a quadratic block relating the segment

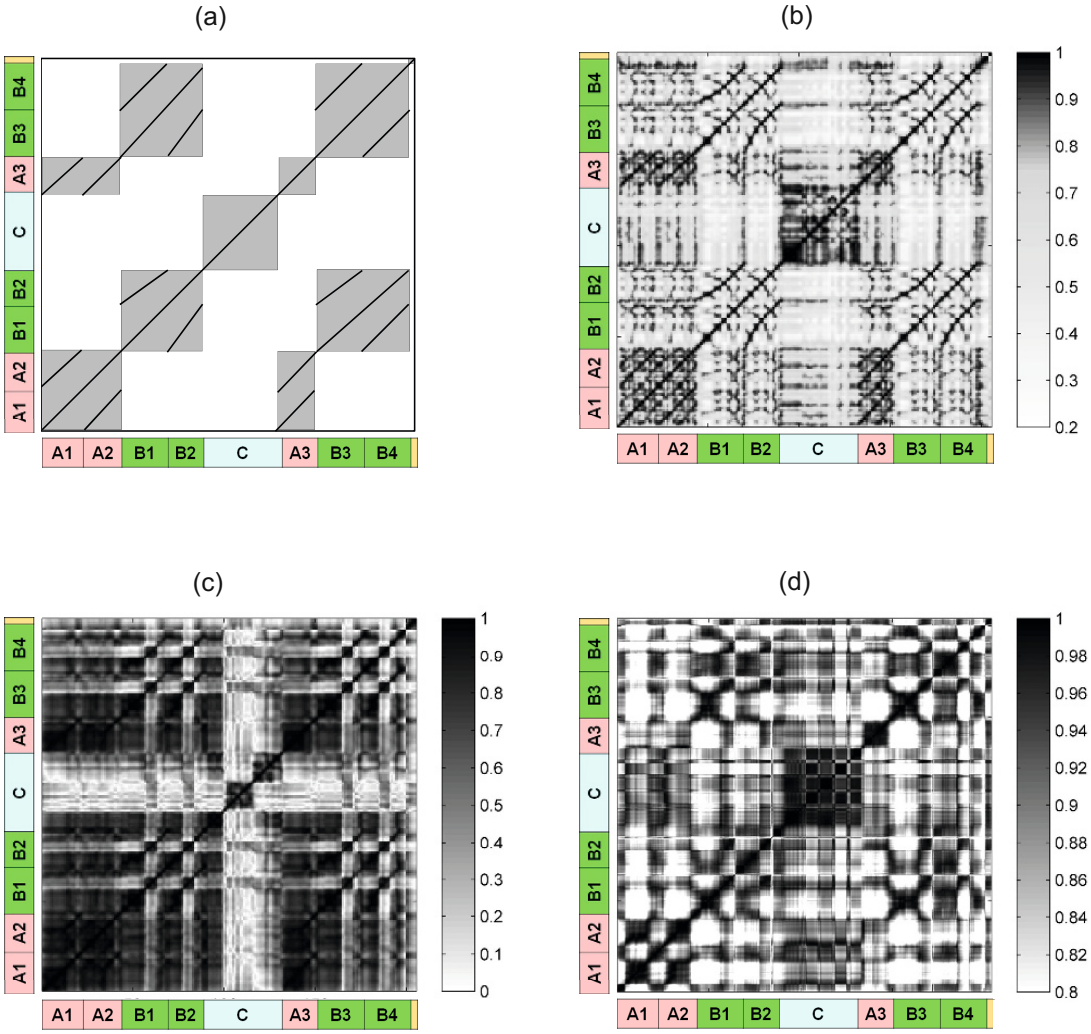


Fig. 4.7 Self-similarity matrices for the Hungarian Dance No. 5 by Johannes Brahms derived from various feature representations shown in Figure 4.6. **(a)** Idealized SSM. **(b)** SSM using chroma-based features. **(c)** SSM using MFCC-based features. **(d)** SSM using tempo-based features.

corresponding to A_1A_2 to itself and another quadratic block relating the A_3-part segment to itself. Furthermore, there are two rectangular blocks, one relating the A_1A_2-part segment to the A_3-part segment and the other relating the A_3-part segment to the A_1A_2-part segment. In case that the three repeating A-part segments are not homogeneous, the SSM reveals path structures that run (more or less) parallel to the main diagonal. For example, there is a path with large similarity values relating A_1 with A_2 and one relating A_1 with A_3.

How are such structures reflected in the case of "real" SSMs? Besides the idealized SSM, Figure 4.7 shows different self-similarity matrices for our Brahms example obtained from the three conceptually different feature sequences of Figure 4.6. In the visualization, large values of $\mathbf{S}$ are indicated by dark gray and small values by light gray. First, one can notice that properties of a self-similarity matrix crucially depend on the respective feature type. The SSM in Figure 4.7b, which is obtained

from chroma-based features, resembles the idealized SSM to a large extent. The block-like structures corresponding to A-part segments indicate that these segments are quite homogeneous with respect to harmony. The same holds for the C-part segment. Furthermore, the small similarity values outside the C-part block (i.e., all cells relating the C-part frames to frames of other segments) show that the C-part segment is harmonically more or less unrelated to all other parts. For the B-part segments, there are path-like structures and no block-like structures. This shows that the B-part segments share the same harmonic progression (i.e., are repetitions with regard to harmony), but are not homogeneous with respect to harmony. An interesting observation is that, even though repeating, the B-part segments are played in different tempi and therefore have different lengths. For example, the shorter B_2-section is played faster than the B_1-section. As a result, the corresponding path does not run exactly parallel to the main diagonal. The gradient of the path indicates the relative tempo difference between the two related segments. Recall that we have discussed a similar issue already in the music synchronization context, where we derived a tempo curve from a warping path (see Section 3.3.2).

Looking at the other two self-similarity matrices the structures are not so clear. The SSM of Figure 4.7c, which results from MFCC-based features, mainly possesses block-like structures. In particular, the C-part segment has a low similarity to all other segments, which indicates a difference in timbre or instrumentation. Now, let us have a look at the tempogram-based SSM shown in Figure 4.7d. Again the C-part segment stands out, thus emphasizing its contrasting role. Furthermore, the SSM indicates the many tempo changes occurring in this music recording. In summary, the musical structure of the Brahms example can be best explained by the repetitive structure of the chroma-based SSM. Since this is the case with many musical works, in particular for melodic and harmonic Western music, we will mainly focus on this type of SSM in the subsequent sections.

We now formalize the concept of paths and blocks (see Figure 4.8). Let $X = (x_1, x_2, \ldots, x_N)$ be a feature sequence and $\mathbf{S}$ the resulting self-similarity matrix. We formally define a segment to be a set $\alpha = [s:t] \subseteq [1:N]$ specified by its starting point s and its end point t (given in terms of feature indices). Let

$$|\alpha| := t - s + 1 \tag{4.4}$$

denote the length of α. Next, a **path** over α of length L is a sequence

$$P = ((n_1, m_1), \ldots, (n_L, m_L)) \tag{4.5}$$

of cells $(n_\ell, m_\ell) \in [1:N]^2$, $\ell \in [1:L]$, satisfying $m_1 = s$ and $m_L = t$ (boundary condition) and $(n_{\ell+1}, m_{\ell+1}) - (n_\ell, m_\ell) \in \Sigma$ (step size condition), where Σ denotes a set of admissible step sizes. Note that this definition is very similar to the one of a warping path (see Section 3.2.1.1). In the case of $\Sigma = \{(1,1)\}$, one obtains paths that are strictly diagonal. In the following, we typically use the set

$$\Sigma = \{(2,1), (1,2), (1,1)\}, \tag{4.6}$$

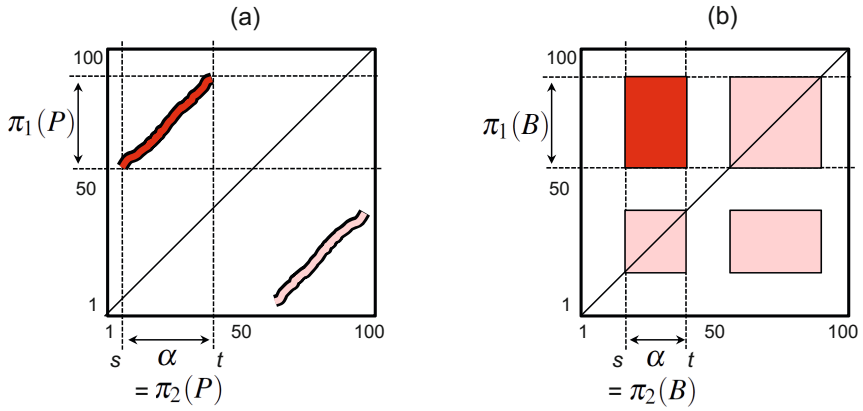


Fig. 4.8 Schematic view of self-similarity matrix with **(a)** a path and **(b)** a block.

which is the step size condition introduced in (3.30). For a path P, one can associate two segments defined by the projections

$$\pi_1(P) := [n_1 : n_L] \quad \text{and} \quad \pi_2(P) := [m_1 : m_L], \tag{4.7}$$

respectively (see Figure 4.8a). The boundary condition enforces $\pi_2(P) = \alpha$. The other segment $\pi_1(P)$ is referred to as the **induced segment**. The **score** $\sigma(P)$ of P is defined as

$$\sigma(P) := \sum_{\ell=1}^{L} \mathbf{S}(n_\ell, m_\ell). \tag{4.8}$$

Note that each path over the segment α encodes a relation between α and an induced segment, where the score $\sigma(P)$ yields a quality measure for this relation.

For blocks, we also introduce corresponding notions. A **block** over a segment $\alpha = [s : t]$ is a subset

$$B = \alpha' \times \alpha \subseteq [1 : N] \times [1 : N] \tag{4.9}$$

for some segment $\alpha' = [s' : t']$. Similar as for a path, we define the two projections $\pi_1(B) = \alpha'$ and $\pi_2(B) = \alpha$ for the block B and call α' the **induced segment** (see Figure 4.8b). Furthermore, we define the score of block B by

$$\sigma(B) = \sum_{(n,m)\in B} \mathbf{S}(n,m). \tag{4.10}$$

Based on paths and blocks, we can now consider different kinds of similarity relations between segments. We say that a segment α_1 is **path-similar** to a segment α_2, if there is a path P of high score with $\pi_1(P) = \alpha_1$ and $\pi_2(P) = \alpha_2$. Similarly, α_1 is **block-similar** to α_2, if there is a block B of high score with $\pi_1(B) = \alpha_1$ and $\pi_2(B) = \alpha_2$. Obviously, in case that the similarity measure s is symmetric, both the self-similarity matrix $\mathbf{S}$ and the above-defined similarity relations between segments are symmetric as well. Another important property of a similarity relation is **transitivity**, i.e., if a segment α_1 is similar to a segment α_2 and segment α_2 is similar

to a segment α_3, then α_1 should also be similar to α_3 (at least to a certain degree). Also this property holds for path- and block-similarity in case that the similarity measure s has this property. As a consequence, path and block structures often appear in groups that fulfill certain symmetry and transitivity properties—at least in the ideal case. For example, if there is a block $B = \alpha' \times \alpha$ of high score, then the symmetry property implies that there is also a block $\alpha \times \alpha'$ of high score. Furthermore, if every frame belonging to α is similar to every other frame of α', then also the frames within the segments α and α' are similar to each other. This leads to additional blocks $\alpha \times \alpha$ and $\alpha' \times \alpha'$ (see Figure 4.8b). Figure 4.7 shows that such groups of similarity relations also appear in "real" SSMs.

Most computational approaches to music structure analysis exploit path- and block-like structures of SSMs in one way or another, and the overall algorithmic pipelines typically contain the following general steps:

1. The music signal is transformed into a suitable feature sequence.
2. A self-similarity matrix is computed from the feature sequence based on a similarity measure.
3. Blocks and paths of high overall score are derived from the SSM. Each block or path defines a pair of similar segments.
4. Entire groups of mutually similar segments are formed from the pairwise relations by applying a clustering step.

The last step can be considered as forming a kind of transitive closure of the pairwise segment relations induced by block and path structures. For example, in the case of Brahms' Hungarian Dance No. 5 (see Figure 4.7), the objective of the last step would be to find one group that contains all A-part segments and another group that contains all B-part segments.

In practice, this general processing pipeline leaves a lot of freedom and needs to be adjusted to account for particular properties of the underlying type of music and the requirements of the intended application. Furthermore, as mentioned before, major challenges arise from the fact that musical parts are rarely repeated in precisely the same way. Instead, audio segments that are considered as repetitions may differ significantly in aspects such as dynamics, orchestration, articulation, tempo, harmony, melody, or any combination of these. As a result, structure analysis becomes a hard and often ill-posed task. In particular, musical and acoustic variations may cause significant deteriorations in the path and block structures and their induced relations. This makes both steps, i.e., the block and path extraction step as well as the grouping step, error-prone and fragile. In the following, we discuss various strategies to cope with such challenges, e.g., by enhancing structural properties of SSMs (Section 4.2.2) or by jointly performing the two error-prone steps of path extraction and grouping within a joint optimization scheme (Section 4.3).

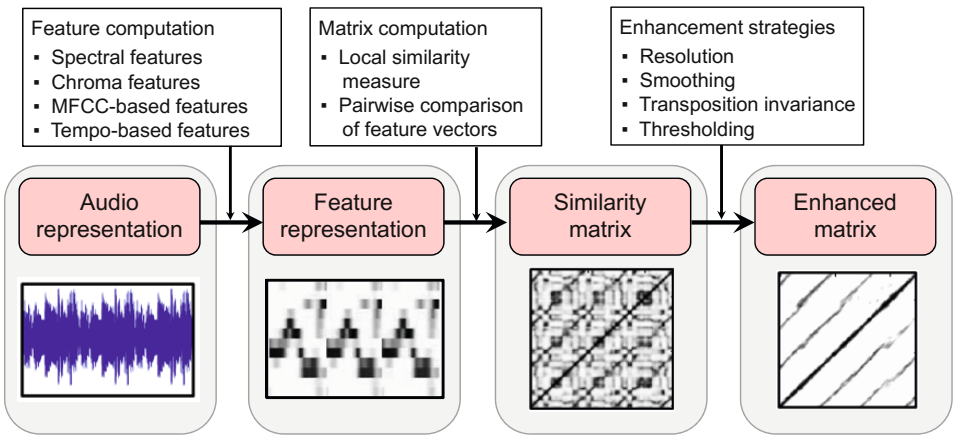


Fig. 4.9 Overview of the similarity matrix computation.

4.2.2 Enhancement Strategies

In this section, we describe various strategies for enhancing structural properties of self-similarity matrices (see Figure 4.9 for an overview). In particular, we focus on augmenting path-like structures, which play a central role in repetition-based structure analysis. Even though all the enhancement strategies are described for self-similarity matrices, similar strategies can be applied for more general similarity or cost matrices.

4.2.2.1 Feature Representation

In the first step, the given waveform-based audio recording is transformed into a suitable feature representation, which captures specific acoustic and musical properties. As we have already discussed in Section 4.2.1 and as illustrated by Figure 4.7, the structural properties of an SSM decisively depend on the feature type used. For example, MFCC-based and related spectral-based features may be suitable to capture aspects such as instrumentation and timbre. Other features based on onset information or tempograms are used to capture beat, tempo, and rhythmic information. In the following, we only consider the case of chroma-based audio features, which relate to harmonic and melodic properties as discussed in Section 3.1.2.

By considering a family of modified chroma representations similar to the ones used in Figure 3.9, we now demonstrate the influence of different parameter settings on the properties of the resulting SSM. Starting with a chroma representation of a given feature rate, this family comes along with two parameters: a length parameter $\ell \in \mathbb{N}$ (given in frames), which is used to smooth or average the feature values over ℓ consecutive frames, as well as a downsampling parameter d, which reduces the feature rate by a factor of d. For a more detailed description of such a procedure, we refer to Section 7.2.1 and Figure 7.10.

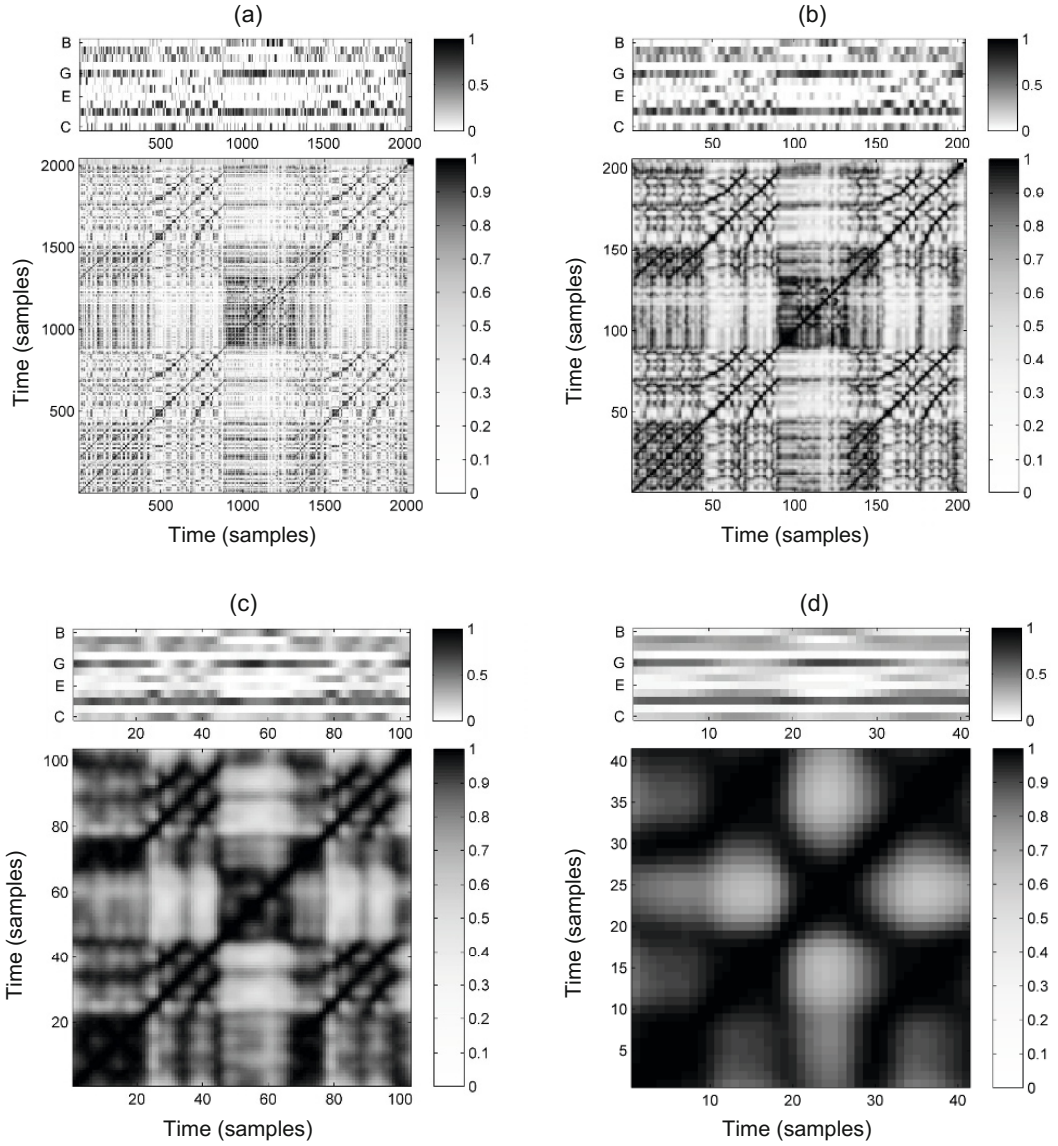


Fig. 4.10 Various chroma representations and resulting SSMs for the Hungarian Dance No. 5 by Johannes Brahms. **(a)** Usage of original normalized chroma features (10 Hz). **(b)** Applying $\ell = 40$ and $d = 10$ (1 Hz). **(c)** Applying $\ell = 160$ and $d = 20$ (0.5 Hz). **(d)** Applying $\ell = 480$ and $d = 50$ (0.2 Hz).

As an example, we start with normalized chroma features with a feature rate of 10 Hz. Figure 4.10a shows the resulting SSM, which yields a very detailed description of repetitive structures. Even though the path structures that correspond to the repeating A-part and B-part segments are visible, the SSM looks quite noisy and many of the shown details are irrelevant when only the overall musical structure is of interest.

Using a smoothing length of $\ell = 40$ (corresponding to four seconds of audio) and a downsampling by $d = 10$ (resulting in a feature rate of 1 Hz), one obtains the SSM shown in Figure 4.10b. Many of the details have been smoothed out, and some of the structurally relevant path and block structures have become more prominent.

In particular, this holds for the paths that relate to the B-part segments. Moreover, reducing the feature rate improves the computational efficiency for subsequent processing steps.

Further increasing the smoothing length and reducing the feature rate results in an emphasis of the rough harmonic content. In particular, neighboring elements in the feature sequence come closer together, which leads to an enhancement of block-like structures. For example, Figure 4.10c shows the SSM when using $\ell = 160$ (16 seconds) and $d = 20$ (feature rate of 0.5 Hz) and Figure 4.10d the SSM using $\ell = 480$ (48 seconds) and $d = 50$ (feature rate of 0.2 Hz). Using large smoothing windows, relevant path structures may be smeared out and lost for the subsequent steps. For other applications such as homogeneity-based structure analysis, however, averaging over large windows may be beneficial.

In summary, this example shows the importance not only of the feature type but also of the size of the analysis window and the feature rate. Knowing the temporal level of the music processing task is of great help for choosing suitable parameters. For example, for tasks such as extracting the musical structure from a given audio recording, smoothing and downsampling already on the feature level can lead to substantial improvements, not to speak of computational benefits in subsequent analysis steps. In particular, running time and memory requirements are important issues when employing concepts such as SSMs, which are quadratic in the length of the input feature sequence. As already mentioned in Section 4.1.3, another important strategy for adjusting and reducing the feature rate is based on **adaptive windowing**, where the analysis windows are determined by previously extracted onset and beat positions. This strategy will be discussed in more detail in Section 6.3.3.

4.2.2.2 Path Smoothing

We have seen that important structural elements of similarity matrices are paths of high similarity that run parallel to the main diagonal. Even though it is often easy for humans to recognize these structures, the automated extraction of paths constitutes a difficult problem due to significant distortions that are caused by variations in parameters such as dynamics, timbre, execution of note groups (e.g., grace notes, trills, arpeggios), modulation, articulation, or tempo progression. As an example, let us have a look at Figure 4.11a, which shows the SSM of a recording of the Waltz No. 2 from Dimitri Shostakovich's Suite for Variety Orchestra No. 1. This piece has the (rough) musical structure $A_1A_2BC_1C_2A_3A_4D$, where the theme, represented by the A-part, appears four times. However, there are significant variations in the four A-parts concerning instrumentation, articulation, as well as dynamics. For example, in A_1 the theme is played by a clarinet, in A_2 by strings, in A_3 by a trombone, and in A_4 by the full orchestra. As is illustrated by Figure 4.11a, these variations result in a rather poor and fragmented path structure. This makes it hard to identify the musically similar segments $\alpha_1 = [4:40]$, $\alpha_2 = [43:78]$, $\alpha_3 = [145:179]$, and $\alpha_4 = [182:217]$ corresponding to A_1, A_2, A_3, and A_4, respectively. In particular, as

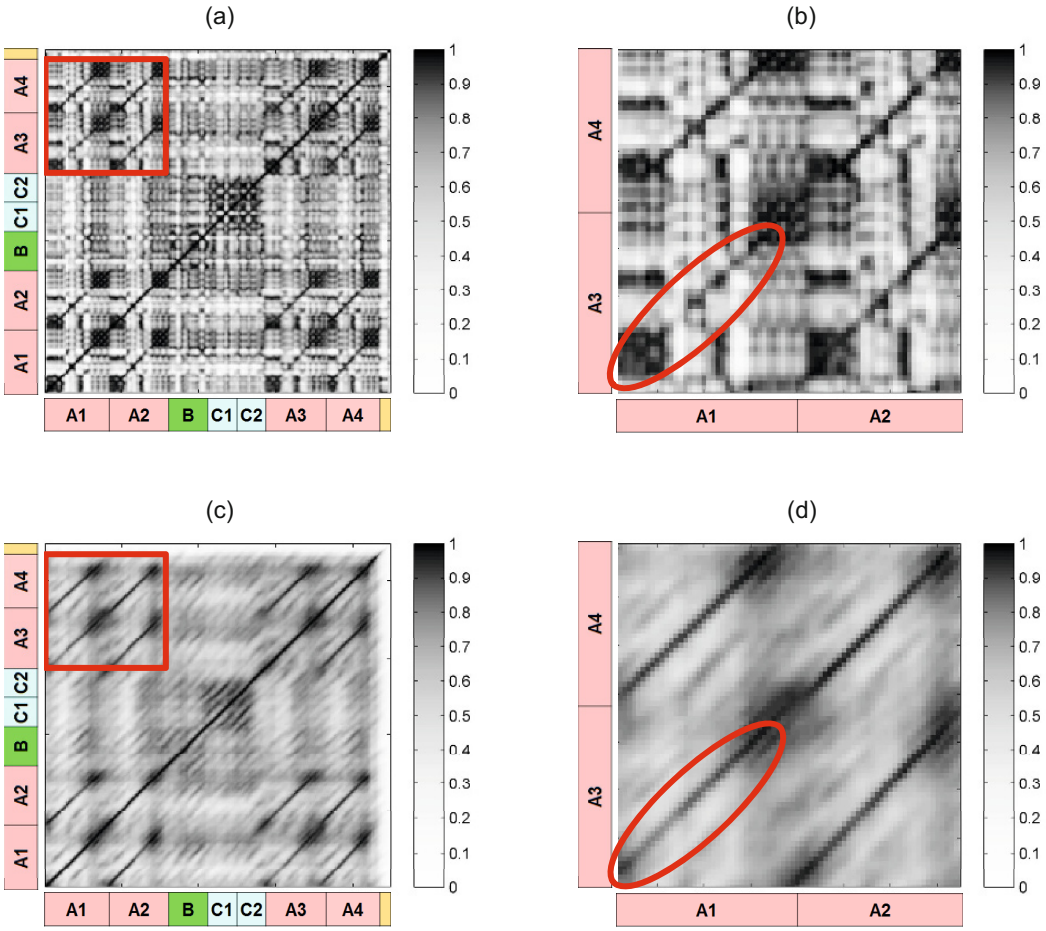


Fig. 4.11 Variants of SSMs for a recording of the Waltz No. 2 from Dimitri Shostakovich's Suite for Variety Orchestra No. 1. **(a)** Original SSM using chroma features (resolution of 1 Hz). **(b)** Enlargement of the submatrix indicated by the rectangular frame in (a). The path corresponding to segments α_1 (part A_1) and α_3 (part A_3) is highlighted by the oval. **(c)** SSM after applying diagonal smoothing. **(d)** Enlargement of the submatrix indicated by the rectangular frame in (c).

can be seen in the enlargement shown in Figure 4.11b, the path corresponding to the segments α_1 and α_3 is quite problematic.

To some extent, as we have seen above, structural properties of the SSM may be augmented by using longer analysis windows in the feature computation step. This, however, may also smooth out important details. As an alternative, we now show how to enhance the path structure of an SSM by applying image processing techniques. Recall that the relevant paths run along the direction of the main diagonal in the case that repeating parts are played in the same tempo. Therefore, in order to augment such paths, the general idea is to apply an averaging filter (or low-pass filter) in the direction of the main diagonal, which results in an emphasis of diagonal information and a softening of other, nondiagonal structures.

We now give a mathematical description of this procedure. Let $\mathbf{S}$ be an SSM of size $N \times N$ and let $L \in \mathbb{N}$ be a length parameter. Then we define the smoothed self-similarity matrix $\mathbf{S}_L$ by setting

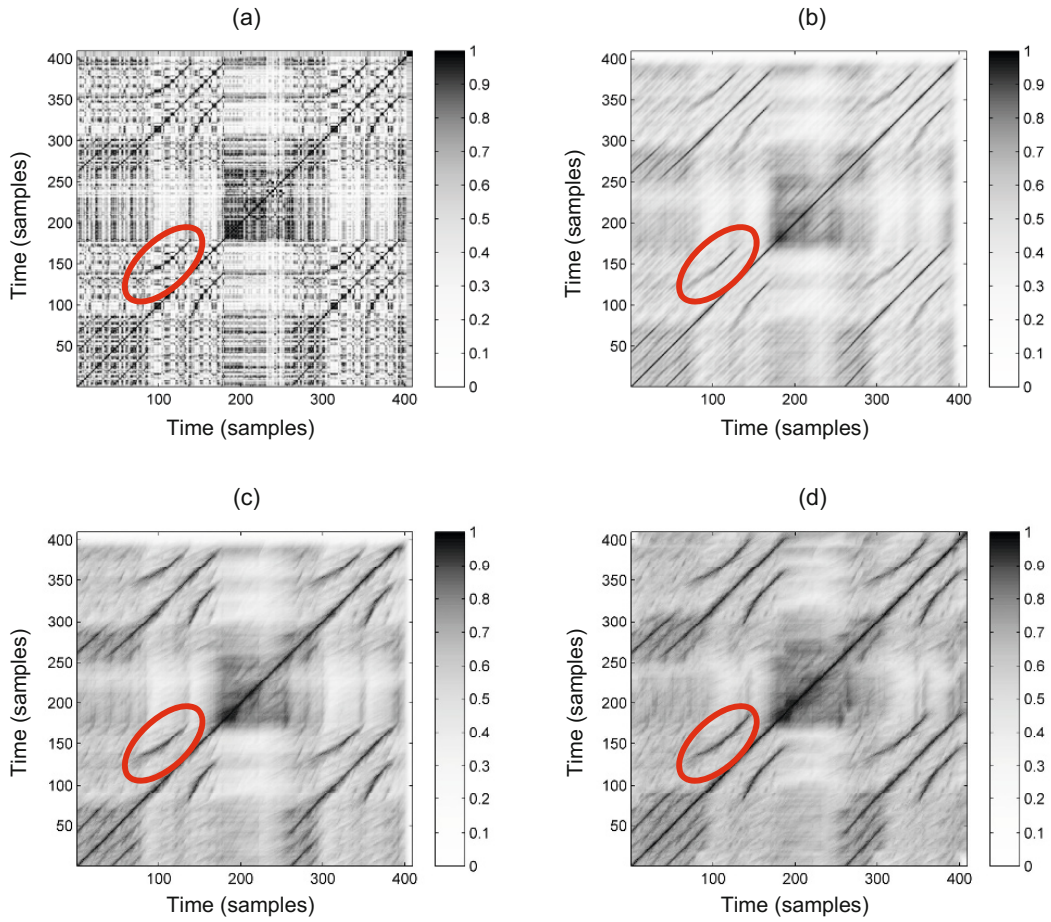


Fig. 4.12 Variants of SSMs for the Hungarian Dance No. 5 by Johannes Brahms. The path corresponding to the B_1-part and B_2-part segments is highlighted. **(a)** Original SSM using chroma features (resolution of 2 Hz). **(b)** SSM after applying diagonal smoothing. **(c)** SSM after applying tempo-invariant smoothing. **(d)** SSM after applying forward–backward smoothing.

$$\mathbf{S}_L(n,m) := \frac{1}{L}\sum_{\ell=0}^{L-1}\mathbf{S}(n+\ell,m+\ell) \tag{4.11}$$

for $n,m \in [1:N-L+1]$. In other words, the value $\mathbf{S}_L(n,m)$ is obtained by averaging the similarity values of two subsequences of length L, one starting at index n and the other at index m. By suitably extending $\mathbf{S}$ (e.g., by **zero-padding** where zero columns and rows are added), we may assume in the following that $\mathbf{S}_L(n,m)$ is defined for $n,m \in [1:N]$.

The averaging procedure results in a smoothing effect along the main diagonal, which is also illustrated by our Shostakovich example of Figure 4.11. Using the length parameter $L = 10$, the resulting self-similarity matrix $\mathbf{S}_{10}$ (Figure 4.11c) reveals the desired path structure much better than the original matrix $\mathbf{S}$ (Figure 4.11a). For example, the enhanced path highlighted in Figure 4.11d reveals the relation between the segments α_1 and α_3 much better than before (see Figure 4.11b).

A simple filtering along the main diagonal only works well if there are no relative tempo differences between the segments to be compared. However, this assumption is violated when a part is repeated with a faster or slower tempo. We have seen such a case in our Brahms example from Figure 4.7, where the shorter B_2-section is played much faster than the B_1-section. It is only the beginning of the B_2-section that is played much faster than the beginning of the B_1-section, whereas the two sections have roughly the same tempo towards the end of the part. This results in a path that does not run exactly parallel to the main diagonal (in particular at the beginning), so that applying an averaging filter in the direction of the main diagonal destroys some of the path structure (see Figure 4.12b). To deal with such relative tempo differences, one idea is to apply a multiple filtering approach, where the SSM is smoothed along various directions that lie in a neighborhood of the direction defined by the main diagonal. Each such direction corresponds to a tempo difference and results in a separate filtered matrix. The final self-similarity matrix is obtained by taking the cell-wise maximum over all these matrices. In this way, the path structure is also enhanced in the presence of local tempo variations as illustrated in Figure 4.12c.

To better understand the details of this procedure, first assume that we have two repeating segments α_1 and α_2 played at the same tempo. Then the direction of the resulting path is given by the gradient $(1,1)$. Next, assume that the second segment α_2 is played at half the tempo compared with α_1. Then the direction of the resulting path is given by the gradient $(1,2)$. In general, if the tempo difference between the two segments is given by a real number $\theta > 0$ (the second segment played θ times slower than the first one), the resulting gradient is $(1,\theta)$. We define the self-similarity matrix smoothed in the direction of $(1,\theta)$ by

$$\mathbf{S}_{L,\theta}(n,m) := \frac{1}{L}\sum_{\ell=0}^{L-1} \mathbf{S}(n+\ell, m+[\ell\cdot\theta]), \tag{4.12}$$

where $[\ell\cdot\theta]$ denotes the integer closest to the real number $\ell\cdot\theta$. Again, by suitably zero-padding the matrix $\mathbf{S}$, we may assume that $\mathbf{S}_{L,\theta}$ is defined for $n,m\in[1:N]$. Now, in practice, one does not know the local tempo difference that may occur in a given music recording. Also, the relative tempo difference between two repeating sections may change over time (as is the case with our Brahms example). Therefore, the idea is to consider a (finite) set Θ consisting of tempo parameters $\theta\in\Theta$ for different relative tempo differences. Then, we compute for each such θ a matrix $\mathbf{S}_{L,\theta}$ and obtain a final matrix $\mathbf{S}_{L,\Theta}$ by a cell-wise maximization over all $\theta\in\Theta$:

$$\mathbf{S}_{L,\Theta}(n,m) := \max_{\theta\in\Theta} \mathbf{S}_{L,\theta}(n,m). \tag{4.13}$$

In practice, one can use prior information on the expected relative tempo differences to determine the set Θ. For example, it rarely happens that the relative tempo difference between repeating segments is larger than 50 percent, so that Θ can be chosen to cover tempo variations of roughly -50 to $+50$ percent. Furthermore, in practice, the tempo range can be covered well by considering only a relatively small number of tempo parameters. For example, a typical choice could be

$\Theta = \{0.66, 0.81, 1.00, 1.22, 1.50\}$ (see Exercise 4.4). Note that choosing $\Theta = \{1\}$ reduces to the case $\mathbf{S}_{L,\Theta} = \mathbf{S}_L$.

This smoothing procedure works in the forward direction, which results in a fading out of the paths, particularly when using a large length parameter. To avoid this fading out, one idea is to additionally apply the averaging filter in a backward direction. The final self-similarity matrix is then obtained by taking the cell-wise maximum over the forward-smoothed and backward-smoothed matrices (see Exercise 4.2). The effect is illustrated in Figure 4.12d by means of the Brahms example.

4.2.2.3 Transposition Invariance

It is often the case that certain musical parts are repeated in a transposed form, where the melody is moved up or down in pitch by a constant interval. As an example, let us consider the song "In the year 2525" by Zager and Evans, which has the musical structure $IV_1V_2V_3V_4V_5V_6V_7BV_8O$. The song starts with a slow intro, which is represented by the I-part. The verse of the song, which is represented by the V-part, is repeated eight times. While the first four verse sections are in the same musical key, V_5 and V_6 are transposed by one semitone upwards, and V_7 and V_8 are transposed by two semitones upwards. Figure 4.13b shows a path-enhanced version of the resulting self-similarity matrix based on some chroma feature representation. This matrix shows path structures that relate the first four V-sections with each other as well as V_5 with V_6 and V_7 with V_8. Because of the transpositions, however, the relation between the first four sections and the last four sections is not reflected in the SSM.

In the following, we show how repetitive structures can be made visible in the SSM even in the presence of key transpositions. We have already seen in Section 3.1.2 that such transpositions can be simulated by cyclically shifting chroma features. Mathematically, we modeled such shifts by the cyclic shift operator $\rho : \mathbb{R}^{12} \to \mathbb{R}^{12}$ defined in (3.11). Now, let $X = (x_1, x_2, \ldots, x_N)$ be the chroma feature sequence. We then define the **i-transposed self-similarity matrix** $\rho^i(\mathbf{S})$ by

$$\rho^i(\mathbf{S})(n,m) := s(\rho^i(x_n), x_m) \tag{4.14}$$

for $n, m \in [1:N]$ and $i \in \mathbb{Z}$. Obviously, one has $\rho^{12}(\mathbf{S}) = \mathbf{S}$. Intuitively, $\rho^i(\mathbf{S})$ describes the similarity relations between the original music recording (represented by $X = (x_1, x_2, \ldots, x_N)$) and the music recording transposed by i semitones upwards (represented by $\rho^i(X) = (\rho^i(x_1), \rho^i(x_2), \ldots, \rho^i(x_N))$). Since one does not know in general the kind of transpositions occurring in the music recording, we apply a similar strategy as before when dealing with relative tempo deviations. Taking a cell-wise maximum over the twelve different cyclic shifts, we obtain a single **transposition-invariant self-similarity matrix** $\mathbf{S}^{\mathrm{TI}}$ defined by

$$\mathbf{S}^{\mathrm{TI}}(n,m) := \max_{i \in [0:11]} \rho^i(\mathbf{S})(n,m). \tag{4.15}$$

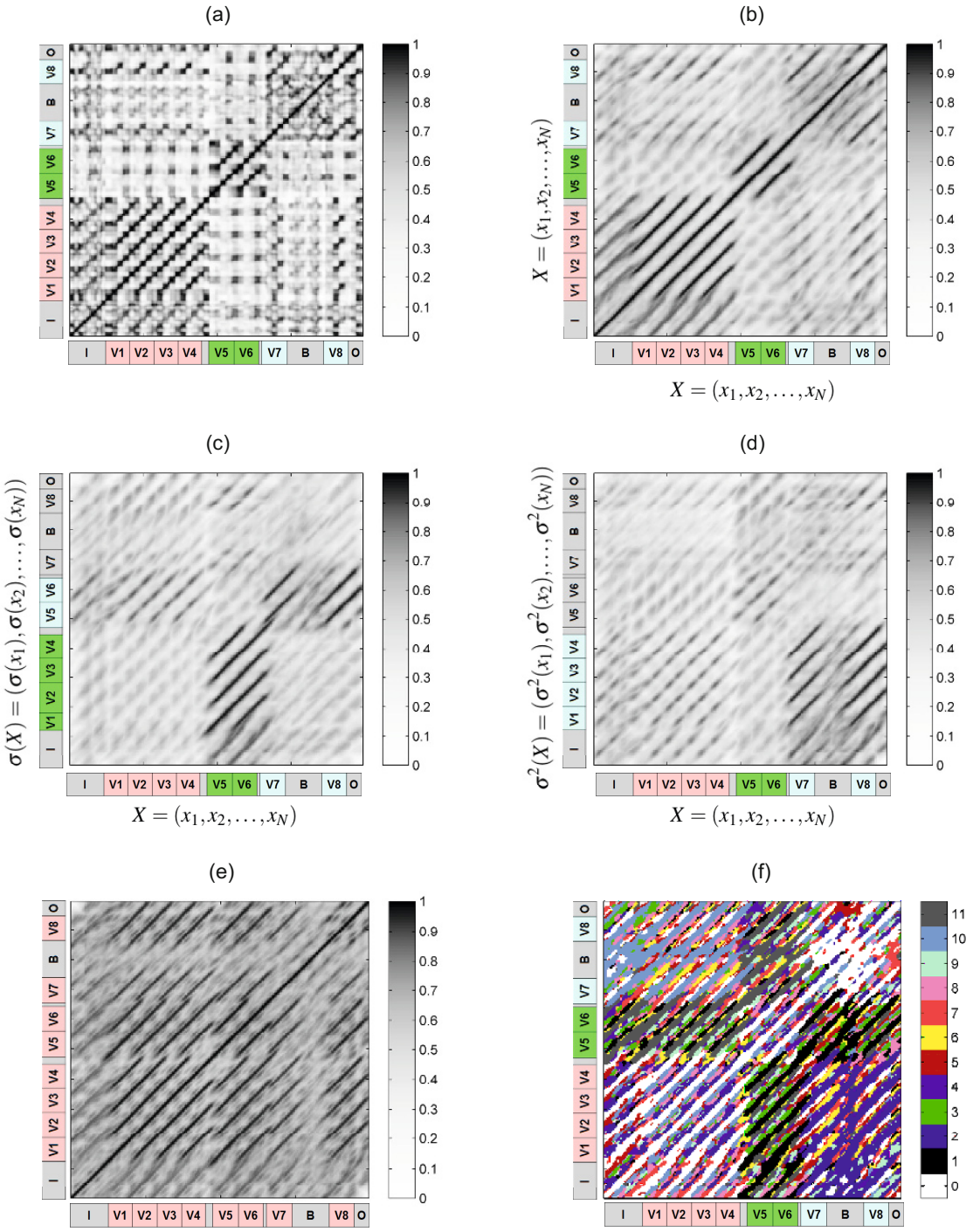


Fig. 4.13 Variants of SSMs for the song "In the year 2525" by Zager and Evans. **(a)** Original SSM using chroma features (resolution of 1 Hz). **(b)** Path-enhanced SSM. **(c)** 1-transposed SSM. **(d)** 2-transposed SSM. **(e)** Transposition-invariant SSM. **(f)** Transposition index matrix.

Furthermore, we store the maximizing shift indices in an additional N-square matrix $\mathbf{I}$, which we refer to as the **transposition index matrix**:

$$\mathbf{I}(n,m) := \underset{i \in [0:11]}{\operatorname{argmax}} \rho^i(\mathbf{S})(n,m). \tag{4.16}$$

We illustrate the definitions by continuing the example shown in Figure 4.13 (see Exercise 4.3). Recall from above that shifting the sections V_1 to V_4 by one semitone upwards makes them similar to the original sections V_5 and V_6. This fact is revealed by the 1-transposed self-similarity matrix shown in Figure 4.13c. Similarly, shifting the sections V_1 to V_4 by two semitones upwards makes them similar to the original sections V_7 and V_8 (see Figure 4.13d). Putting together the information of all i-transposed self-similarity matrices by the maximization in (4.15), one obtains the transposition-invariant self-similarity matrix $\mathbf{S}^{\mathrm{TI}}$ shown in Figure 4.13e, where all pairwise similarity relations between the eight V-part segments become visible.

The resulting transposition index matrix is shown in Figure 4.13f in a color-coded form. We first discuss the case that the matrix $\mathbf{I}$ assumes the value $i = 0$ (white color in Figure 4.13f). The value $i = 0$ for a cell (n,m) indicates that $s(\rho^i(x_n), x_m)$ assumes a maximal value for $i = 0$. In other words, the chroma vector x_m is closer to x_n than to any other shifted version of x_n. Note, however, that this does not necessarily mean that x_m is close to x_n in absolute terms. As may be expected, the maximizing index is $i = 0$ at all positions where the conventional self-similarity matrix shown in Figure 4.13b reveals paths of low cost. Next, we consider the case that the matrix $\mathbf{I}$ assumes the value $i = 1$ (black color in Figure 4.13f). The value $i = 1$ for a cell (n,m) indicates that x_n becomes most similar to x_m when shifted one semitone upwards. Thus the strong path relations shown in Figure 4.13c correspond to cells assuming the value $i = 1$, and so on.

At this point, we want to note that introducing transposition invariance by cell-wise maximization over several matrices may increase the noise level in the resulting similarity matrix. Therefore, the transposition-invariant matrix should be computed on the basis of smoothed matrices, since the smoothing typically goes along with a suppression of unwanted noise. The definitions in (4.14) and (4.15) can be easily combined with the averaging approaches described by (4.11) and (4.12) to yield matrices $\rho^i_{L,\Theta}(\mathbf{S})$ and $\mathbf{S}^{\mathrm{TI}}_{L,\Theta}$. Such matrices are shown in Figure 4.13.

4.2.2.4 Thresholding

In many music analysis applications, self-similarity matrices are further processed by suppressing all values that fall below a given threshold. On the one hand, such a step often leads to a substantial reduction of unwanted noise-like components while leaving only the most significant structures. On the other hand, weaker but still relevant information may be lost. The thresholding strategy used may have a significant impact on the final result and has to be carefully chosen in the context of the considered application. Figure 4.14 shows some examples obtained by different thresholding settings as explained below.

The simplest strategy is to apply **global thresholding**, where all values $\mathbf{S}(n,m)$ of a similarity matrix $\mathbf{S}$ below a given threshold parameter $\tau > 0$ are set to zero:

$$\mathbf{S}_\tau(n,m) := \begin{cases} \mathbf{S}(n,m) & \text{if } \mathbf{S}(n,m) \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{4.17}$$

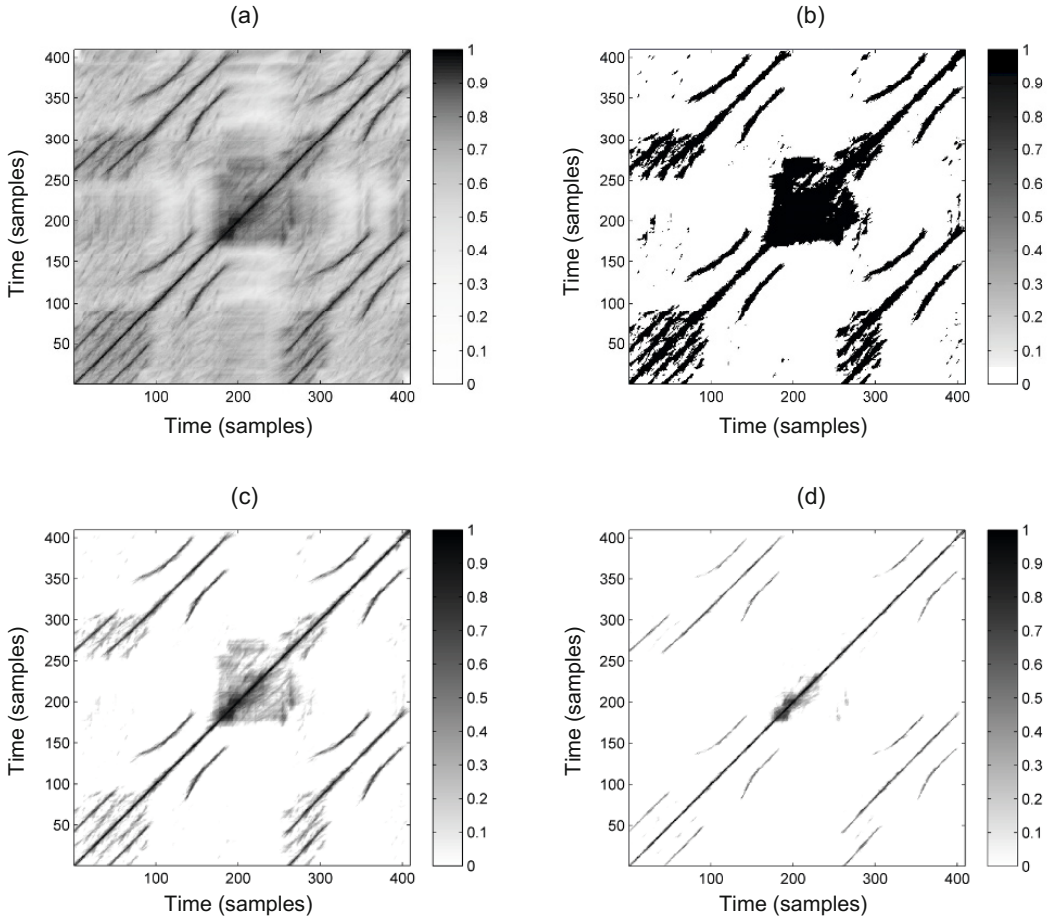


Fig. 4.14 Thresholding strategies applied to an SSM for the Hungarian Dance No. 5 by Johannes Brahms. **(a)** SSM from Figure 4.12d. **(b)** SSM after thresholding and binarization ($\tau = 0.75$). **(c)** SSM after thresholding and scaling ($\rho = 0.2$). **(d)** SSM after thresholding and scaling ($\rho = 0.05$).

Also, binarization of the similarity matrix can be applied by setting all values above or equal to the threshold to one and all others to zero. Instead of binarization, one may perform a scaling where the range $[\tau, \mu]$ is linearly scaled to $[0,1]$ in the case that $\mu := \max_{n,m}\{\mathbf{S}(n,m)\} > \tau$, otherwise all entries are set to zero. Sometimes it may be beneficial to introduce an additional penalty parameter $\delta \leq 0$, setting all original values below the threshold to the value δ (see Section 4.3 for an application of this variant).

The global threshold τ can also be chosen in a **relative** fashion by keeping $\rho \cdot 100\%$ of the cells with the highest values using a relative threshold parameter $\rho \in [0,1]$. Finally, thresholding can also be performed using a more **local** strategy by thresholding in a column- and rowwise fashion. To this end, for each cell (n,m), the value $\mathbf{S}(n,m)$ is kept if it is among the $\rho \cdot 100\%$ of the largest cells in row n and at the same time among the $\rho \cdot 100\%$ of the largest cells in column m, all other values being set to zero (see Exercise 4.5). As said before, the suitability of a thresholding setting depends on the respective music material and the application in mind. Often, suitable thresholds are learned and optimized using supervised learning procedures.

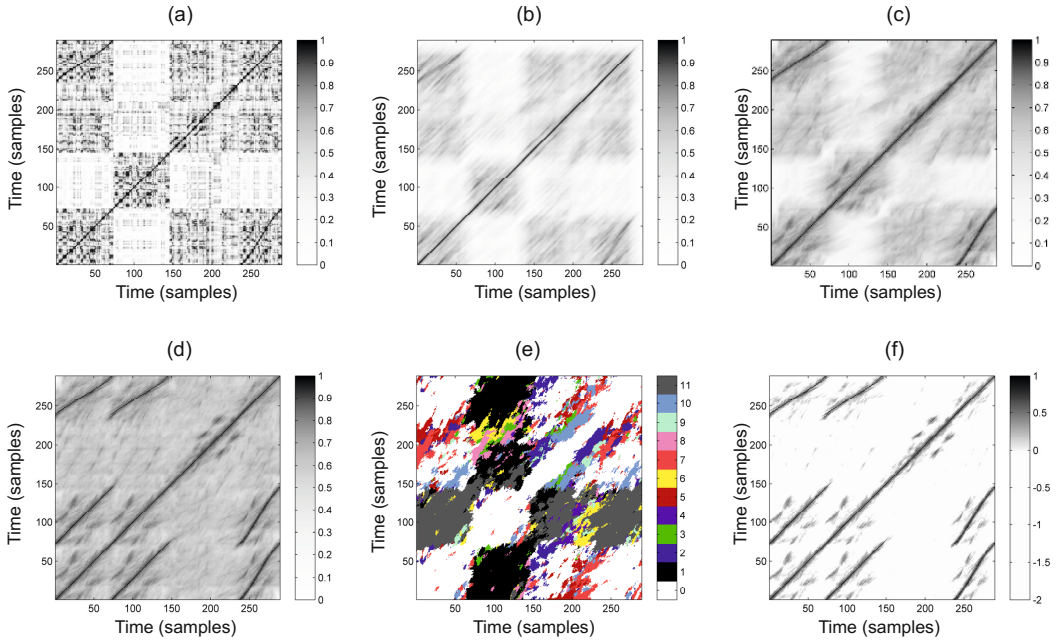


Fig. 4.15 Variants of similarity matrices for the same audio recording. **(a)** Original SSM using chroma features of 2 Hz resolution. **(b)** SSM after applying diagonal smoothing. **(c)** SSM after applying tempo-invariant and forward–backward smoothing. **(d)** Transposition-invariant SSM. **(e)** Transposition index matrix. **(f)** SSM after thresholding with penalty and scaling ($\rho = 0.2$, $\delta = -2$).

To conclude this section, Figure 4.15 summarizes the various enhancement and processing steps applied to a music recording having the musical structure $A_1A_2BA_3$. In this example, A_2 is a modulation of A_1 transposed by one semitone upwards, whereas A_3 is a repetition of A_1, however played much faster. Figure 4.15 shows a typical processing pipeline for computing an SSM as used in structure analysis applications. First, the music recording is converted into a sequence of normalized and smoothed chroma features as in Figure 3.9. Then, based on the similarity measure (4.3), an enhanced transposition-invariant self-similarity matrix $\mathbf{S}^{\mathrm{TI}}_{L,\Theta}$ is computed (see Figure 4.15d). In the next step, global thresholding is applied using a threshold parameter τ and a penalty parameter δ. Furthermore, the range $[\tau, 1]$ is linearly scaled to $[0, 1]$. As a result, the relevant path structure tends to lie in the positive part of the resulting SSM, whereas all other cells are given a negative score. Finally, setting $\mathbf{S}(n,n) = 1$ for $n \in [1:N]$, one can introduce a normalization property, which may have been lost in the smoothing process due to boundary effects. The SSM shown in Figure 4.15f is obtained in this way using a feature rate of 2 Hz. Settings for the enhancement are $L = 20$ for the length parameter and $\Theta = \{0.50, 0.63, 0.79, 1.26, 1.59, 2.00\}$ for the set of relative tempo differences (see Exercise 4.4). In this example, the threshold is chosen in a relative fashion by using the relative threshold $\rho = 0.2$ and the penalty parameter is set to $\delta = -2$.

4.3 Audio Thumbnailing

In this section, we deal with a prominent subproblem of music structure analysis commonly known as **audio thumbnailing**. Given a music recording, the objective is to automatically determine the most representative section, which may serve as a kind of "preview" giving a listener a first impression of the song or piece of music. Based on such previews, the user should be able to quickly decide if he or she would like to listen to the song or to move on to the next recording. Thus, audio thumbnails are an important browsing and navigation aid for finding interesting pieces in large music collections.

Often sections such as the chorus or the main theme of a song are good candidates for audio thumbnails. Such parts are typically repeated several times throughout the recording. Therefore, to determine a thumbnail automatically, most procedures try to identify a section that has on the one hand a certain minimal duration and on the other many (approximate) repetitions. As we have seen before, one challenge is that such repeating sections may show significant acoustic and musical differences in aspects that concern dynamics, instrumentation, articulation, and tempo.

We now describe a typical thumbnailing procedure for extracting the most repetitive segment from a given music recording. In particular, we show how enhanced self-similarity matrices as well as time warping techniques are applied for dealing with multiple variabilities. As the main technical tool, we introduce in Section 4.3.1 a fitness measure that assigns a fitness value to each audio segment. This measure simultaneously captures two aspects. First, it indicates **how well** a given segment explains other related segments, and second, it indicates **how much** of the overall music recording is covered by all these related segments. The audio thumbnail is then defined to be the segment of maximal fitness. In the computation of the fitness measure, one important concept is to avoid hard decisions and error-prone steps in an early stage of the algorithmic pipeline. To this end, an optimization scheme is applied for jointly performing path extraction and grouping—two error-prone steps that are often performed successively. Furthermore, we also have a look at an efficient algorithm based on dynamic programming for computing the fitness measure. In Section 4.3.2, we then introduce the concept of a scape plot representation that shows the fitness values over all possible audio segments. A visualization of this fitness scape plot yields a compact high-level view on the structural properties of the entire music recording. Finally, in Section 4.3.3, we discuss several explicit examples to indicate the potential as well as the limitations of the presented thumbnailing approach.

4.3.1 Fitness Measure

The idea of the fitness measure to be introduced is to simultaneously establish all relations between a given segment and its repetitions. To this end, a self-similarity matrix is required as described at the end of Section 4.2.2 and illustrated

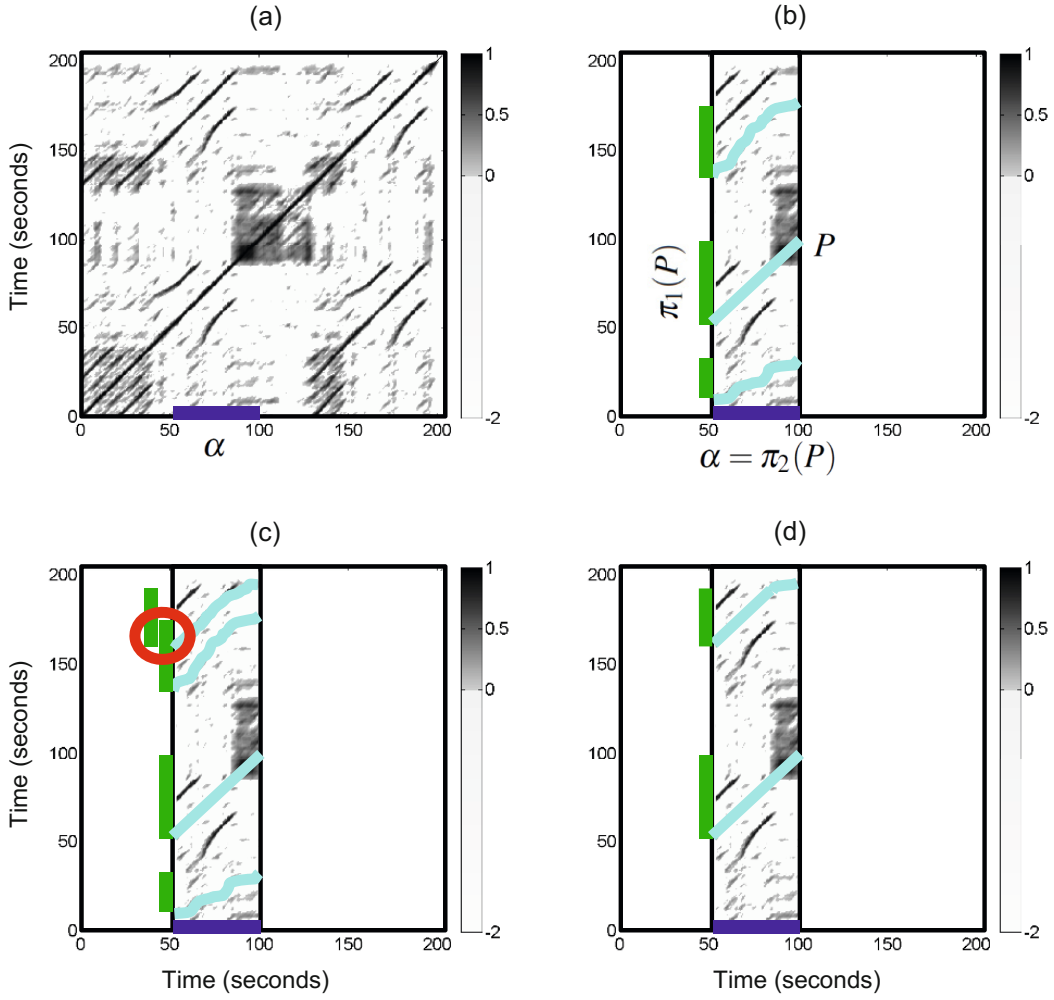


Fig. 4.16 SSM of our Brahms example with various paths over the segment $\alpha = [50:100]$. The induced segments are indicated on the vertical axis. **(a)** SSM. **(b)** Paths forming a path family. **(c)** Paths not forming a path family (induced segments overlap). **(d)** Paths forming an optimal path family.

by Figure 4.15f. Our Brahms example (see Figure 4.16) will serve as a running example for the subsequent steps. The following description of the fitness measure is generic in the sense that it works with general self-similarity matrices that only fulfill some basic normalization properties. From a technical point of view, only the properties

$$\mathbf{S}(n,m) \leq 1 \tag{4.18}$$

for all $n,m \in [1:N]$ and

$$\mathbf{S}(n,n) = 1 \tag{4.19}$$

for all $n \in [1:N]$ are required.

4.3.1.1 Path Family

Recall from Section 4.2.1 that a path P over a given segment $\alpha = [s:t] \subseteq [1:N]$ encodes a relation between $\alpha = \pi_2(P)$ and the induced segment $\pi_1(P)$. The score $\sigma(P)$ defined in (4.8) yields a quality measure for this relation. Extending the notion of a path, we now introduce the concept of a path family, which allows us to capture relations between α and several other segments in the music recording. To this end, we first define a **segment family** of size K to be a set

$$\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_K\} \tag{4.20}$$

of pairwise disjoint segments, i.e., $\alpha_k \cap \alpha_j = \emptyset$ for all $k, j \in [1:K]$ with $k \neq j$. Let

$$\gamma(\mathcal{A}) := \sum_{k=1}^{K} |\alpha_k| \tag{4.21}$$

be the **coverage** of $\mathcal{A}$ (see (4.4)). A **path family** over α is defined to be a set

$$\mathcal{P} := \{P_1, P_2, \ldots, P_K\} \tag{4.22}$$

of size K, consisting of paths P_k over α for $k \in [1:K]$. Furthermore, as an additional condition, we require that the induced segments are pairwise disjoint. In other words, the set $\{\pi_1(P_1), \ldots, \pi_1(P_K)\}$ is required to be a segment family. This definition is illustrated by Figure 4.16b, which shows a path family over the segment $\alpha = [50:100]$ consisting of $K = 3$ paths P_1, P_2, and P_3. The induced segments are $\pi_1(P_1) = [10:35]$, $\pi_1(P_2) = [50:100]$, and $\pi_1(P_3) = [136:174]$, which are pairwise disjoint. In contrast, the example shown in Figure 4.16c does not yield a path family, since the disjointness condition of the induced segments is violated. Extending the definition in (4.8), the **score** $\sigma(\mathcal{P})$ of the path family $\mathcal{P}$ is defined as

$$\sigma(\mathcal{P}) := \sum_{k=1}^{K} \sigma(P_k). \tag{4.23}$$

As indicated by Figure 4.16, there are in general a large number of possible path families over α. Among these path families, let

$$\mathcal{P}^* := \operatorname*{argmax}_{\mathcal{P}} \sigma(\mathcal{P}) \tag{4.24}$$

denote an optimal path family of maximal score (see Figure 4.16d for an example). In the following, the family consisting of the segments induced by the paths of $\mathcal{P}^*$ will be referred to as the **induced segment family** (of $\mathcal{P}^*$ or of α). Intuitively, the induced segment family contains the (nonoverlapping) repetitions of the segment α. Next, we show how an optimal path family $\mathcal{P}^*$ can be computed efficiently using dynamic programming and then explain how the fitness measure is derived from the score $\sigma(\mathcal{P}^*)$ and the induced segment family of $\mathcal{P}^*$.

4.3.1.2 Optimization Scheme

We now describe an efficient algorithm for computing an optimal path family for a given segment in a running time that is linear in the product of the length of the segment and the length of the entire music recording. The algorithm is based on a modification of dynamic time warping (DTW) as discussed in Section 3.2. Recall that, given two sequences, say $X = (x_1, x_2, \ldots, x_N)$ and $Y = (y_1, y_2, \ldots, y_M)$, the objective of DTW is to compute an optimal path that **globally** aligns X and Y, where the first elements as well as the last elements of the two sequences are to be aligned. The step size condition as specified by the set Σ constrains the slope of the path. In particular, using $\Sigma = \{(2,1),(1,2),(1,1)\}$, as specified in (3.30) and (4.6), each element of X is aligned to at most one element of Y (and vice versa).

Now, when computing an optimal path family over a given segment $\alpha = [s:t] \subseteq [1:N]$, the role of Y is taken over by the segment α, and the conditions change compared with classical DTW. In particular, α can be simultaneously aligned to several (nonoverlapping) subsequences of X. However, for each such subsequence, the entire segment α is to be aligned. Furthermore, certain sections of X may be left completely unconsidered in the alignment. Finally, instead of finding a **cost-minimizing** warping path, we are now looking for a **score-maximizing** path family. To account for the new constraints, we need to introduce additional steps that allow us to skip certain sections of X and to jump from the end to the beginning of the given segment α. The following procedure is also illustrated by Figure 4.17.

First, considering paths over the segment $\alpha = [s:t]$ with $M := |\alpha|$, we only consider the $N \times M$ submatrix $\mathbf{S}^\alpha$, which consists of the columns s to t of the self-similarity matrix $\mathbf{S}$. Next, we specify an accumulated score matrix $\mathbf{D} \in \mathbb{R}^{N,M+1}$ by a recursive procedure (similar to (3.25) for the accumulated cost matrix). The rows of $\mathbf{D}$ are indexed by $[1:N]$, and the columns are indexed by $[0:M]$, where the role of the column indexed by $m = 0$ is explained later. For a given cell (n,m), we consider a **set of predecessors** denoted by $\Phi(n,m)$, which contains all cells that may precede (n,m) in a valid path family. For $n \in [2:N]$ and $m \in [2:M]$ this set is given by

$$\Phi(n,m) = \{(n-i, m-j) \mid (i,j) \in \Sigma\} \cap [1:N] \times [1:M], \tag{4.25}$$

and the accumulated score matrix is defined by

$$\mathbf{D}(n,m) = \mathbf{S}^\alpha(n,m) + \max\{\mathbf{D}(i,j) \mid (i,j) \in \Phi(n,m)\}. \tag{4.26}$$

So far, this is similar to the recursion of the DTW algorithm summarized in Table 3.2. The constraint conditions and additional steps are realized by the definition of the values of $\mathbf{D}$ for the remaining index pairs (n,m) with $n = 1$ or $m \in \{0,1\}$.

As said before, the first column of $\mathbf{D}$ indexed by $m = 0$ plays a special role. We define this first column recursively by $\mathbf{D}(1,0) = 0$ and

$$\mathbf{D}(n,0) = \max\{\mathbf{D}(n-1,0), \mathbf{D}(n-1,M)\} \tag{4.27}$$

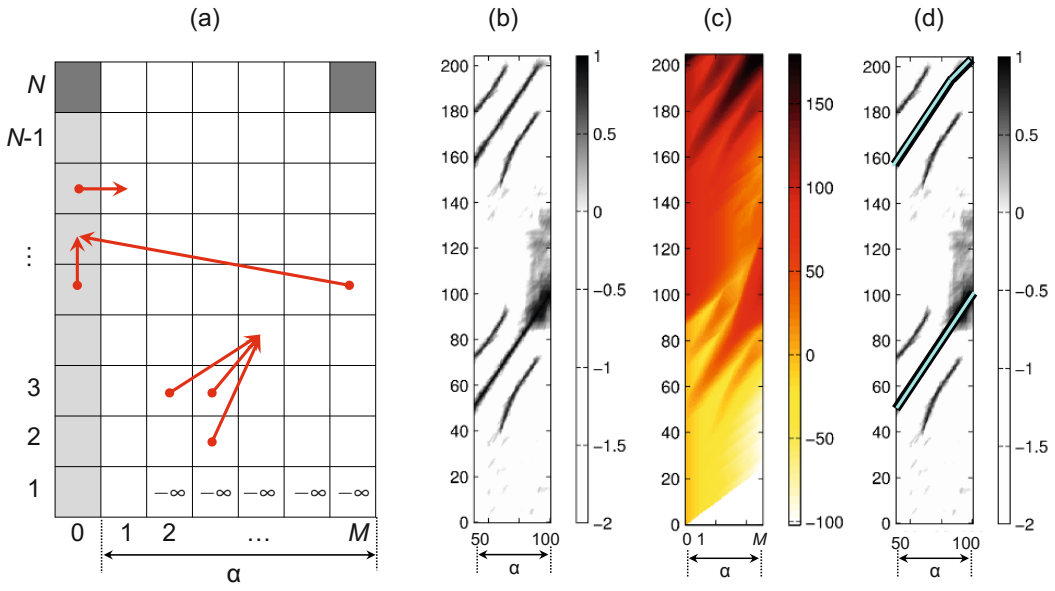


Fig. 4.17 **(a)** Illustration of the various predecessors in computing the accumulated score matrix. **(b)** Submatrix $\mathbf{S}^{\alpha}$ with $\alpha = [50:100]$ of the SSM shown in Figure 4.16a. **(c)** Accumulated score matrix $\mathbf{D}$. **(d)** Optimal path family.

for $n \in [2:N]$. The first term $\mathbf{D}(n-1,0)$ enables the algorithm to move upwards without accumulating any (possibly negative) score, thus realizing the condition that sections of X may be skipped without penalty (negative score). The second term $\mathbf{D}(n-1,M)$ closes up a path (ensuring that the entire segment α is aligned to a subsequence of X), while ensuring that the next possible segment does not overlap with the previous segment. Intuitively, the column indexed by $m=0$ may be thought of as a kind of "elevator" column that makes it possible to skip arbitrary sections of X and to initialize new path components.

Next, we define the second column of $\mathbf{D}$ indexed by $m=1$ by

$$\mathbf{D}(n,1) = \mathbf{D}(n,0) + \mathbf{S}^{\alpha}(n,1) \tag{4.28}$$

for $n \in [1:N]$. This definition makes it possible to start a new path component at cell $(n,1)$ coming from any position of the "elevator" column. Finally, to complete the initialization, we set $\mathbf{D}(1,m) = -\infty$ for $m \in [2:M]$. This forces the first path to come from the elevator column, thus starting with the first element of α. The score of an optimal path family is then given by

$$\sigma(\mathcal{P}^*) = \max\{\mathbf{D}(N,0), \mathbf{D}(N,M)\}. \tag{4.29}$$

The first term $\mathbf{D}(N,0)$ reflects the case that the final section of X may be skipped, and the second term $\mathbf{D}(N,M)$ ensures that in the other case the entire segment α is aligned to a suffix of X. The associated optimal path family $\mathcal{P}^*$ can be constructed from $\mathbf{D}$ using a backtracking algorithm as in the DTW algorithm (see Table 3.2). As the only modification, the cells of $\mathbf{S}^{\alpha}$ that belong to the first auxiliary column (in-

dexed by $m = 0$) are to be omitted to obtain the final path family. In Exercise 4.6, we show that the recursive procedure for computing $\mathbf{D}$ has a computational complexity (in terms of memory requirements and running time) of $O(MN)$.

4.3.1.3 Definition of Fitness Measure

We have seen how to efficiently compute for a given segment α an optimal path family $\mathcal{P}^* = \{P_1, \ldots, P_K\}$, which reveals the repetition relations of α. In view of our intended fitness measure, one first idea is to simply use the total score $\sigma(\mathcal{P}^*)$ as defined in (4.23) as the fitness value for α. However, this measure does not yet have the desired properties, since it not only depends on the lengths of α and the paths, but also captures trivial self-explanations. For example, the segment $\alpha = [1:N]$ explains the entire sequence X perfectly, which is a trivial fact. More generally, each segment α explains itself perfectly, information that is encoded by the main diagonal of a self-similarity matrix. Therefore, one idea in defining the fitness measure is to disregard such trivial self-explanations. Assuming the normalization properties (4.18) and (4.19) of the underlying self-similarity matrix $\mathbf{S}$, this step can be done by simply subtracting the length $|\alpha|$ from the score $\sigma(\mathcal{P}^*)$. For example, in the case $\alpha = [1:N]$ this leads to the value zero. Furthermore, we normalize the score with regard to the lengths $L_k := |P_k|$ of the paths P_k contained in the optimal path family $\mathcal{P}^*$. This yields the **normalized score** $\bar{\sigma}(\alpha)$ defined by

$$\bar{\sigma}(\alpha) := \frac{\sigma(\mathcal{P}^*) - |\alpha|}{\sum_{k=1}^{K} L_k}. \tag{4.30}$$

From the assumption $\mathbf{S}(n,n) = 1$, we obtain $\bar{\sigma}(\alpha) \geq 0$ (see Exercise 4.7). Furthermore, note that, when using $\Sigma = \{(1,2),(2,1),(1,1)\}$, we get $\sum_k L_k \leq N$. This together with $\mathbf{S}(n,m) \leq 1$ implies the property $\bar{\sigma}(\alpha) \leq 1 - |\alpha|/N$. Intuitively, the value $\bar{\sigma}(\alpha)$ expresses the **average score** of the optimal path family $\mathcal{P}^*$ (minus a proportion for the self-explanation).

The normalized score indicates **how well** a given segment explains other segments, where the normalization eliminates the influence of segment lengths. This makes the normalized score a fair measure when comparing segments of different lengths. Besides repetitiveness, another issue is **how much** of the underlying music recording is covered by the thumbnail and its related segments. To capture this property, we define a **coverage** measure for a given α. To this end, let $\mathcal{A}^* := \{\pi_1(P_1), \ldots, \pi_1(P_K)\}$ be the segment family induced by the optimal path family $\mathcal{P}^*$, and let $\gamma(\mathcal{A}^*)$ be its coverage as defined in (4.21). Similar to the normalized score, we define the **normalized coverage** $\bar{\gamma}(\alpha)$ by

$$\bar{\gamma}(\alpha) := \frac{\gamma(\mathcal{A}^*) - |\alpha|}{N}. \tag{4.31}$$

As above, the length $|\alpha|$ is subtracted to compensate for trivial coverage. Obviously, one has $\bar{\gamma}(\alpha) \leq 1 - |\alpha|/N$. In other words, the value $\bar{\gamma}(\alpha)$ expresses the ratio be-

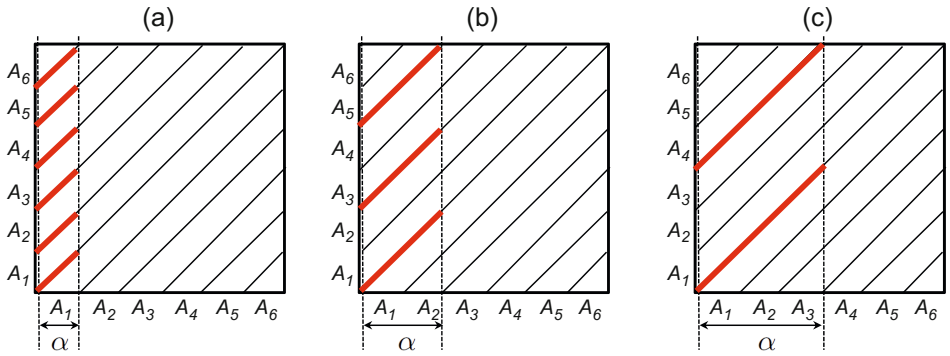


Fig. 4.18 Idealized SSM corresponding to the musical structure $A_1A_2\ldots A_6$ with optimal path families for various segments α corresponding to **(a)** A_1, **(b)** A_1A_2, and **(c)** $A_1A_2A_3$.

tween the union of the induced segments of α and the total length of the original recording (minus a proportion for the self-explanation).

Having a high average score and a high coverage are both desirable properties for defining a thumbnail segment. However, these two properties are sometimes hard to satisfy at the same time. Shorter segments often have a higher average score, but a lower coverage, whereas longer segments tend to have a lower average score, but a higher coverage. To balance out these two trends, we combine the score and coverage measure by taking a suitable average. There are many ways for combining two values including the arithmetic, the geometric, and the harmonic mean. In the following, we use the harmonic mean, which (compared with the arithmetic mean) tends towards the smaller element and mitigates the impact of large differences between the two numbers to be averaged (see Exercise 4.8). We define the **fitness** $\varphi(\alpha)$ of the segment α to be the **harmonic mean**

$$\varphi(\alpha) := 2 \cdot \frac{\bar{\sigma}(\alpha) \cdot \bar{\gamma}(\alpha)}{\bar{\sigma}(\alpha) + \bar{\gamma}(\alpha)} \tag{4.32}$$

between the normalized score and normalized coverage. The fitness measure inherits the property $\varphi(\alpha) \leq 1 - |\alpha|/N$ from $\bar{\sigma}(\alpha)$ and $\bar{\gamma}(\alpha)$. The effect of combining score and coverage is illustrated by Figure 4.21 and will be further discussed in Section 4.3.2.

As an example, Figure 4.18 shows an idealized SSM of a piece having the musical structure $A_1A_2\ldots A_6$, where we assume that each part is played in exactly the same way. Furthermore, we assume that the SSM has the value one on the indicated paths and otherwise the value zero. Let us first consider the segment α corresponding to A_1. The optimal path family consists of six paths over α (see Figure 4.18a). Since trivial self-explanations are left unconsidered, one obtains a normalized score of $\bar{\sigma}(\alpha) = 5/6$ and a normalized coverage of $\bar{\gamma}(\alpha) = 5/6$, which results in a fitness of $\varphi(\alpha) = 5/6$. Similarly, one obtains $\varphi(\alpha) = 2/3$ for the segment α corresponding to A_1A_2 (Figure 4.18b), and $\varphi(\alpha) = 1/2$ for the segment α corresponding to $A_1A_2A_3$ (Figure 4.18c). Obviously, the fitness is $\varphi(\alpha) = 0$ in case α corresponds to the entire music recording. In conclusion, the fitness measure allows for comparing segments

of different length while slightly favoring shorter segments (since self-explanations are neglected). Further examples are discussed in Section 4.3.3 (see Exercise 4.9).

4.3.1.4 Thumbnail Selection

Based on the fitness measure, we define the audio thumbnail to be the segment of maximal fitness:

$$\alpha^* := \operatorname*{argmax}_{\alpha} \varphi(\alpha). \tag{4.33}$$

By construction of the fitness measure, this segment has nonoverlapping repetitions that cover a possibly large portion of the audio recording. Furthermore, these repetitions are given by the induced segments obtained by the optimal path family of α^* yielding a segmentation of the audio recording into pairwise disjoint segments.

To account for prior knowledge and to remove spurious estimates, one can impose additional requirements on the thumbnail solution. In particular, introducing a lower bound θ for the minimal possible thumbnail length allows us to reduce the effect of noise scattered in the underlying self-similarity matrix. Extending the above definition, we define

$$\alpha^*_\theta := \operatorname*{argmax}_{\alpha, |\alpha| \geq \theta} \varphi(\alpha). \tag{4.34}$$

In the next sections, we discuss and illustrate the properties of the fitness measure and the thumbnailing procedure in more detail.

4.3.2 Scape Plot Representation

The fitness measure assigns to each possible segment a fitness value that expresses a certain property. We now introduce a representation by which this segment-dependent property can be visualized in a compact and hierarchical way. Recall that a segment $\alpha = [s:t] \subseteq [1:N]$ is uniquely determined by its starting point s and its end point t. Since any two numbers $s,t \in [1:N]$ with $s \leq t$ define a segment, there are $(N+1)N/2$ different segments (see Exercise 4.10). Now, instead of considering start and end points, each segment can also be uniquely described by its center

$$c(\alpha) := (s+t)/2 \tag{4.35}$$

and its length $|\alpha|$. Using the center to parameterize a horizontal axis and the length to parameterize the height, each segment can be represented by a point in a triangular representation (see Figure 4.19). This way, the set of segments are ordered from bottom to top in a hierarchical way according to their length. In particular, the top of this triangle corresponds to the unique segment of maximal length N and the bottom points of the triangle correspond to the N segments of length one (where the start point coincides with the end point). Furthermore, all segments $\alpha' \subseteq \alpha$ contained in

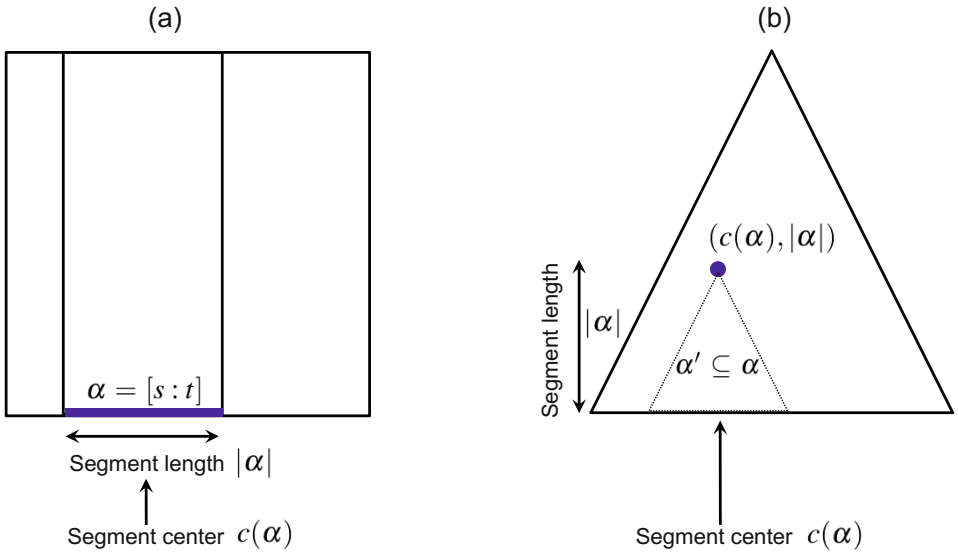


Fig. 4.19 Definition of scape plot representation. **(a)** Schematic SSM with segment. **(b)** Schematic scape plot with segment.

a given segment α correspond to points in the triangular representation that lie in a subtriangle below the point given by α (see Figure 4.19b and Exercise 4.12).

The triangular representation can be used as a grid for indicating the fitness values of all segments, which we also refer to as a **scape plot** representation of the fitness measure. More precisely, we define a scape plot Δ by setting

$$\Delta(c(\alpha), |\alpha|) := \varphi(\alpha) \tag{4.36}$$

for segment α. For our Brahms example, Figure 4.20 shows a scape plot representation in a color-coded form. Note that the maximal entry of Δ corresponds to the maximal fitness value, thus defining the thumbnail α^*.

4.3.3 Discussion of Properties

We now discuss some examples to illustrate the properties of the introduced fitness measure, the scape plot representation, and the induced segmentation. In our first example, we continue with our Brahms example. Recall that this piece has the musical structure $A_1A_2B_1B_2CA_3B_3B_4D$ (see Figure 4.5). Figure 4.16a shows a self-similarity matrix obtained from a given audio recording of this piece. Based on this SSM, the fitness measure is evaluated for all segments. The resulting fitness scape plot, which is shown in Figure 4.20a, reflects the musical structure in a hierarchical way. First note that the fitness-maximizing segment is $\alpha^* = [68:89]$. The coordinates in the scape plot are specified by the center $c(\alpha) = 78.5$ and the length $|\alpha| = 22$. Musically, this segment corresponds to the B_2-part, which is indeed the most repetitive part. The induced segment family consists of the four B-

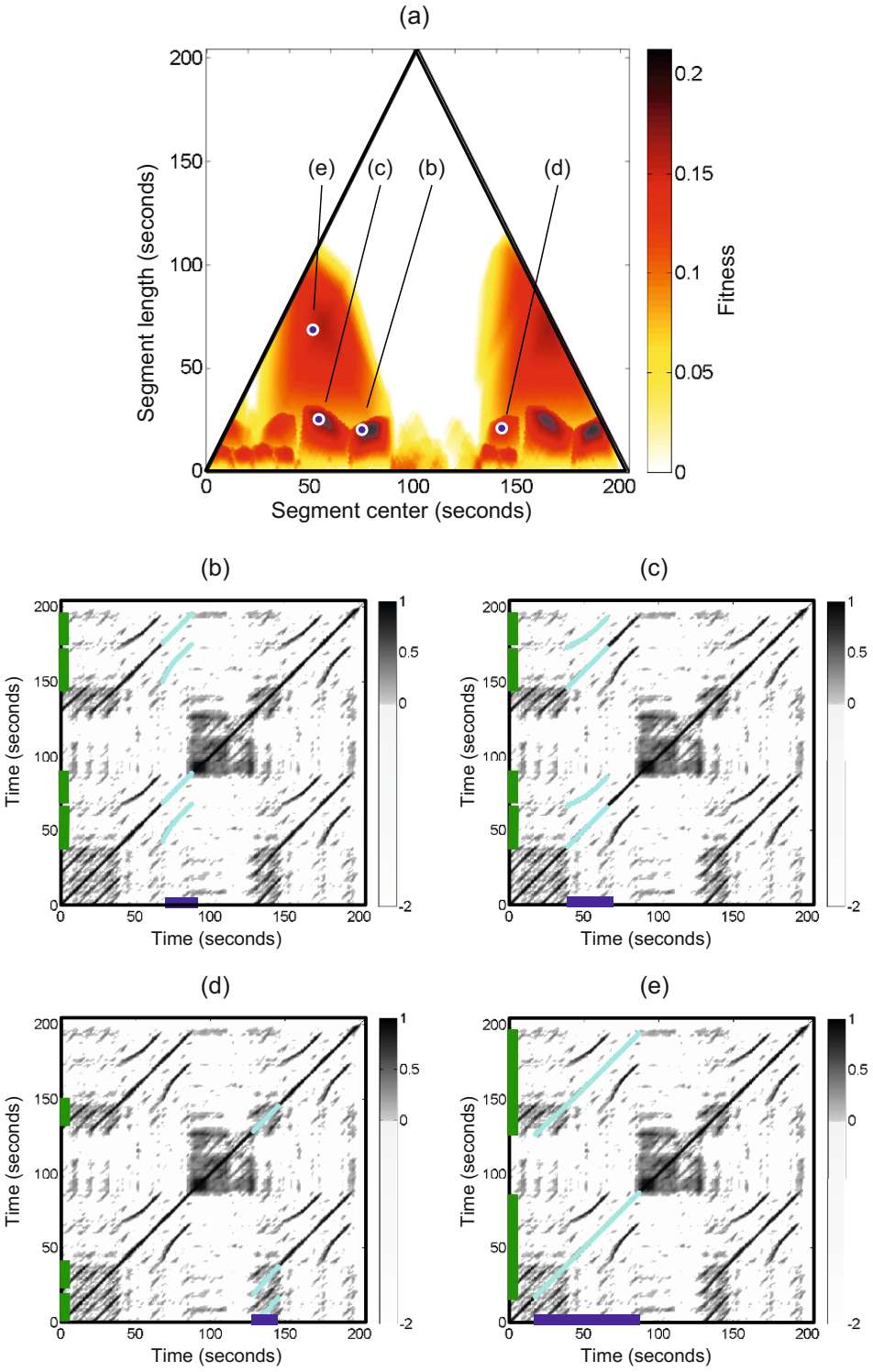


Fig. 4.20 Scape plot representation of fitness measure as well as different optimal path families and induced segment families over different segments α for our Brahms example. **(a)** Fitness scape plot. **(b)** $\alpha = \alpha^* = [68:89]$ (the thumbnail segment of maximal fitness corresponding to B_2). **(c)** $\alpha = [41:67]$ (corresponding to B_1). **(d)** $\alpha = [131:150]$ (corresponding to A_3). **(e)** $\alpha = [21:89]$ (corresponding to $A_1B_1B_2$).

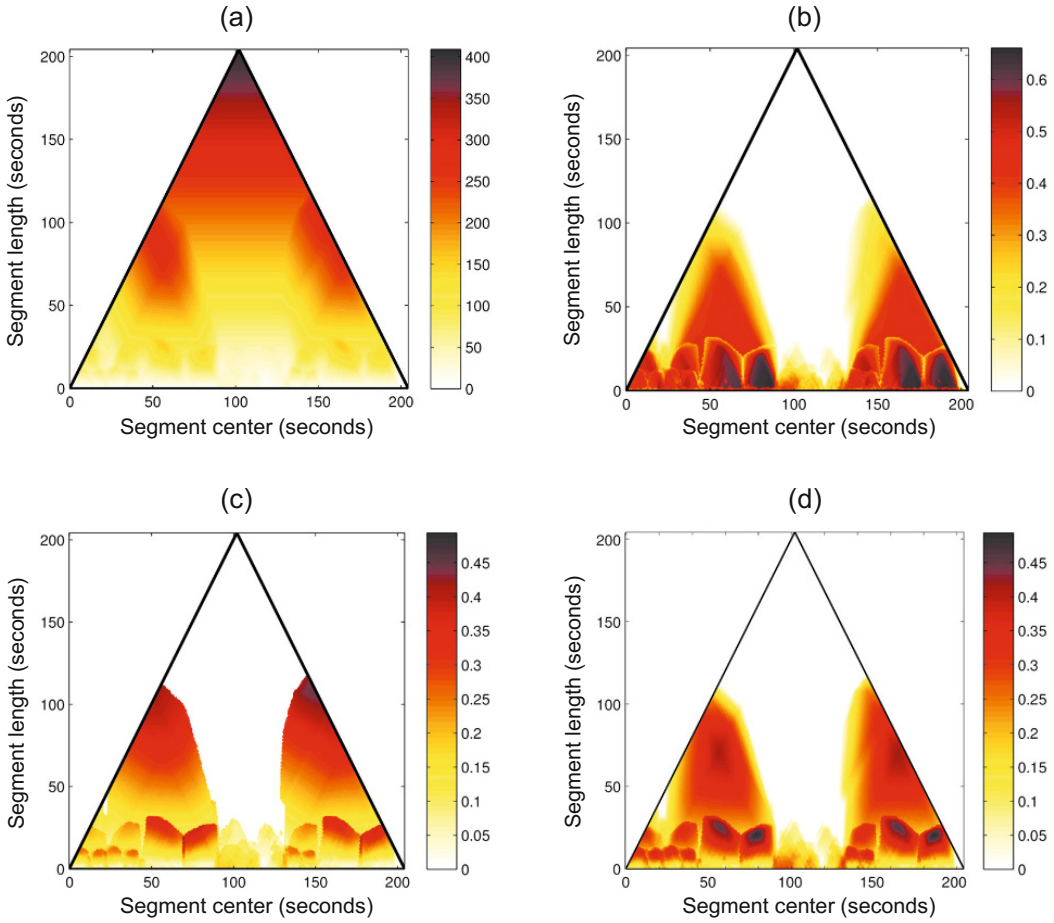


Fig. 4.21 Various scape plot representations. **(a)** Score. **(b)** Normalized score. **(c)** Normalized coverage. **(d)** Fitness measure (harmonic mean of (b) and (c)).

part segments (see Figure 4.20b). Note that all four B-part segments have almost the same fitness and lead to more or less the same segment family. For example, Figure 4.20c shows the induced segment family obtained from the B_1-part segment. This reflects the fact that each of the B-part segments may serve equally well as the thumbnail.

Recall that the introduced fitness measure slightly favors shorter segments (see Exercise 4.9). Therefore, since in this recording the B_2-part is played faster than the B_1-part, the fitness measure favors the B_2-part segment over the B_1-part segment. The scape plot also reveals other local maxima of musical relevance. For example, the local maximum corresponding to segment $\alpha = [131 : 150]$ ($c(\alpha) = 140.5$, $|\alpha| = 20$) corresponds to the A_3-part, and the induced segment family reveals the three A-parts (see Figure 4.20d). Furthermore, the local maximum assumed for segment $\alpha = [21 : 89]$ ($c(\alpha) = 55$, $|\alpha| = 69$) corresponds to $A_2B_1B_2$, which is repeated as $A_3B_3B_4$ (see Figure 4.20e). Again, note that, because of the normalization where self-explanations are disregarded, the fitness of the rather long segment $\alpha = [21 : 89]$ is well below that of the thumbnail $\alpha^* = [68 : 89]$.

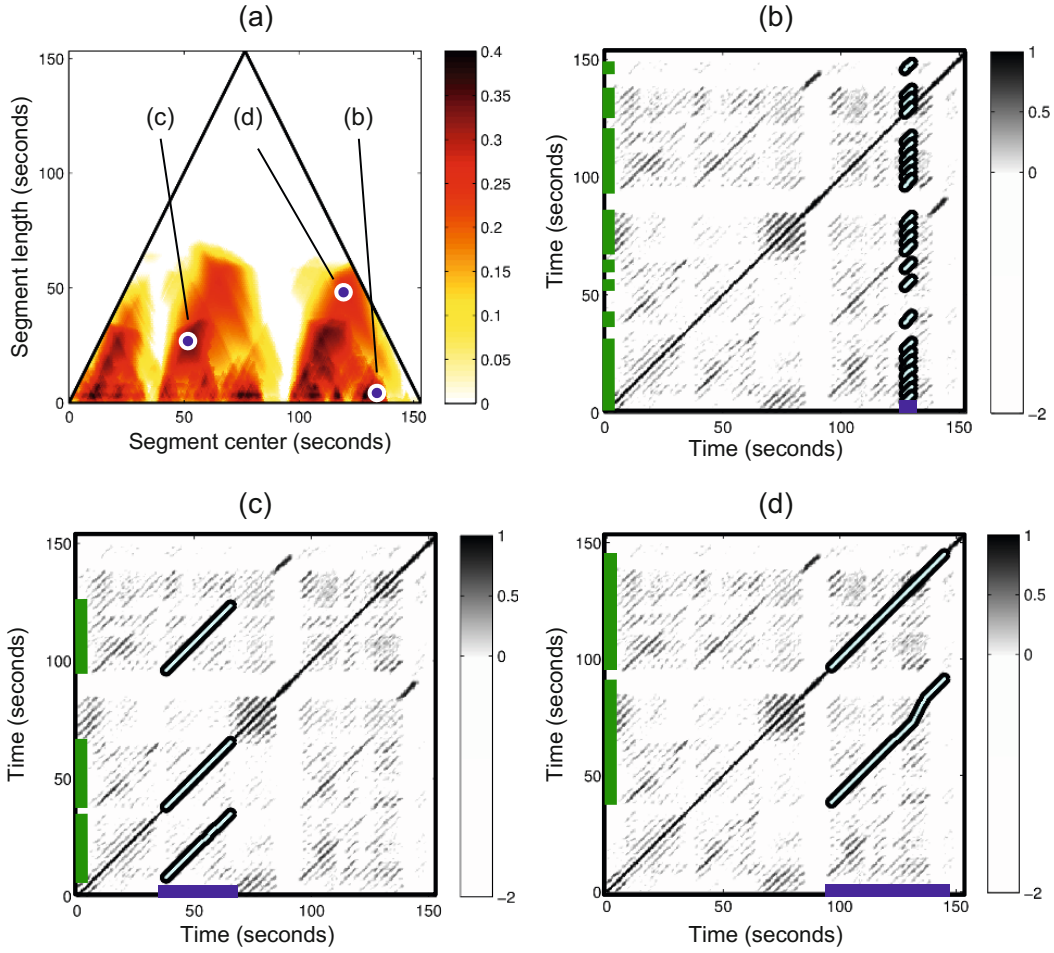


Fig. 4.22 Various optimal path families and induced segment families over different segments α for the Beatles song "Twist and Shout" having the musical structure $IV_1V_2B_1V_3B_2O$. **(a)** Fitness scape plot. **(b)** $\alpha = \alpha^* = [127:130]$. **(c)** $\alpha = \alpha^*_\theta = [38:65]$ using $\theta = 10$ (corresponding to V_2). **(d)** $\alpha = \alpha^*_\theta = [97:145]$ using $\theta = 40$ (corresponding to V_3B_2).

Next, we illustrate that in the definition (4.32) of the fitness measure the combination of the normalized score (4.30) and coverage (4.31) is of crucial importance. Figure 4.21b shows the scape plot when only using the normalized score. Since this measure expresses the average score of a path family without expressing how much of the audio material is actually covered, many of the small segments have a relatively high score. Using such a measure would typically result in false-positive segments of small length. In contrast, using only the normalized coverage would typically favor longer segments (see Figure 4.21c). The corresponding path families often contain components of rather low overall score (just above zero), which may result in rather weak repetitions. By combining score and coverage, the fitness measure balances out the two conflicting principles of having strong repetitions (high score) and of explaining possibly large portions of the recording (high coverage). Finally, we illustrate the importance of the normalization step by looking at the score $\sigma(\mathcal{P}^*)$ of the optimizing path family $\mathcal{P}^*$ over a segment α (see (4.23)). Figure 4.21d shows the resulting scape plot representation. Without normalization,

longer segments typically dominate the shorter segments, with the entire recording having maximal score.

As a second example, Figure 4.22 shows the scape plot and various induced segment families for the Beatles song "Twist and Shout." This song has the musical structure $IV_1V_2B_1V_3B_2O$ consisting of a short intro (I-part), three verses (V-part), two bridges (B-part), and an outro (O-part). Interestingly, the fitness-maximizing segment $\alpha^* = [127 : 130]$ is very short and leads to a large number of spurious induced segments (see Figure 4.22b). The reason is that the song contains a short harmonic phrase, a so-called **riff**, which is repeated over and over again. As a consequence, the self-similarity matrix contains many repeated spurious path fragments which, as a whole family, lead to a high score as well as to a high coverage. To circumvent such problems, one can consider the segment α^*_θ as defined in (4.34) to enforce a minimal length for the thumbnail. In our example, setting $\theta = 10$ (given in seconds) one obtains the segment $\alpha^*_\theta = [38 : 65]$, which corresponds to the verse V_2 (see Figure 4.22c). This indeed yields a musically meaningful thumbnail. By further increasing the lower bound, one obtains superordinate repeating parts such as $\alpha^*_\theta = [97 : 145]$ corresponding to V_3B_2 (when using $\theta = 40$) (see Figure 4.22d).

4.4 Novelty-Based Segmentation

While the audio thumbnailing approach described in the previous section was based on the principle of repetition, we now discuss some segmentation procedures that are based on the principle of novelty. Recall from Section 4.1.1 that segment boundaries are often accompanied by a change in instrumentation, dynamics, harmony, tempo, or some other characteristics. It is the objective of novelty-based structure analysis to locate points in time where such musical changes occur, thus marking the transition between two subsequent structural parts. There are numerous approaches for novelty detection described in the literature. In the following, we present the main ideas of two of these approaches while introducing some general concepts that are also useful for the analysis of general time series. We start with a classical procedure where local changes are detected by correlating a checkerboard-like kernel along the main diagonal of a self-similarity matrix (Section 4.4.1). This procedure works particularly well when the underlying SSM has block-like structures. Then we introduce an approach for novelty detection that is based on structure features that encapsulate both local and global properties of the audio recording (Section 4.4.2). This procedure also highlights how various segmentation principles can be applied jointly within a single segmentation framework.

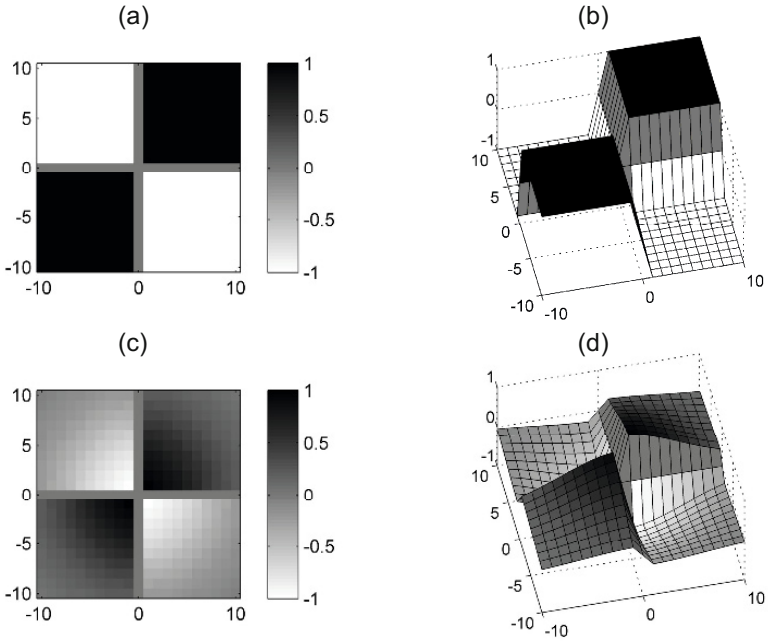


Fig. 4.23 Checkerboard kernel functions of size $M = 21$ ($L = 10$). **(a,b)** Box-like checkerboard kernel and 3D plot. **(c,d)** Gaussian checkerboard kernel and 3D plot.

4.4.1 Novelty Detection

As we have seen in Section 4.2.1, a self-similarity matrix reveals block-like structures in the case that the underlying feature sequence stays somewhat constant over the duration of an entire section. Often such a homogeneous segment is followed by another homogeneous segment that stands in contrast to the previous one. For example, a section played by strings may be followed by a section played by brass. Or there may be two contrasting sections each being homogeneous with respect to harmony, where the boundary between these sections is characterized by a change in the musical key. We have encountered such a case in our Brahms example, where one has homogeneous A-part segments in G minor and homogeneous C-part segments in G major (Figure 4.5).

One idea in novelty detection is to identify the boundary between two homogeneous but contrasting segments by correlating a checkerboard-like kernel function along the main diagonal of the SSM. This yields a **novelty function**. The peaks in this function indicate instances where significant changes occur in the audio signal. For example, using MFCCs, these peaks are good indicators for changes in timbre or instrumentation. Similarly, using chroma-based features, one obtains indicators for changes in harmony.

We now explain this procedure in more detail. As before, let $X = (x_1, x_2, \ldots x_N)$ be a feature sequence and $\mathbf{S}$ a self-similarity matrix of size $N \times N$ derived from X. Let us first consider an audio recording that consists of two homogeneous but contrasting sections. When visualized, the resulting SSM looks like a 2×2 checker-

board as shown in Figure 4.23a. The two dark blocks on the main diagonal correspond to the regions of high similarity within the two sections. In contrast, the light regions outside these blocks express that there is a low cross-similarity between the sections. Thus, to find the boundary between the two sections one needs to identify the crux of the checkerboard. This can be done by correlating $\mathbf{S}$ with a kernel that itself looks like a checkerboard. The simplest such kernel is the (2×2)-unit kernel defined by

$$\mathbf{K} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.37}$$

This kernel can be written as the difference between a "coherence" and an "anti-coherence" kernel. The first kernel measures the self-similarity on either side of the center point and will be high when each of the two regions is homogeneous. The second kernel measures the cross-similarity between the two regions and will be high when there is little difference across the center point. The difference between the two values estimates the **novelty** of the feature sequence at the center point. The novelty is high when the two regions are self-similar but different from each other.

In audio structure analysis, where one is typically interested in changes on a larger time scale, kernels of larger size are used. Furthermore, since in this book we adopt a centered view (where a physical time position is associated to the center of a window or kernel), we assume that the size of the kernel is odd given by $M = 2L+1$ for some $L \in \mathbb{N}$. A box-like checkerboard kernel $\mathbf{K}_{\text{Box}}$ of size M is an $(M \times M)$ matrix, which is indexed by $[-L:L] \times [-L:L]$. The matrix is defined by

$$\mathbf{K}_{\text{Box}} = \operatorname{sgn}(k) \cdot \operatorname{sgn}(\ell), \tag{4.38}$$

where $k, \ell \in [-L:L]$ and "sgn" is the sign function (being -1 for negative numbers, 0 for zero, and 1 for positive numbers). For example, in the case $L = 2$, one obtains

$$\mathbf{K}_{\text{Box}} = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \end{bmatrix} \tag{4.39}$$

(see Figure 4.23a). Note that the zero row and the zero column in the middle have been introduced more for theoretical reasons to ensure the symmetry of the kernel matrix. The checkerboard kernel can be smoothed to avoid edge effects using windows that taper towards zero at the edges. For this purpose, one may use a radially symmetric Gaussian function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$\phi(s,t) = \exp(-\varepsilon^2(s^2+t^2)), \tag{4.40}$$

where the parameter $\varepsilon > 0$ allows for adjusting the degree of tapering. Then the kernel $\mathbf{K}_{\text{Gauss}}$ tapered by the Gaussian function is given by pointwise multiplication:

$$\mathbf{K}_{\text{Gauss}}(k,\ell) = \phi(k,\ell) \cdot \mathbf{K}_{\text{Box}}(k,\ell), \tag{4.41}$$

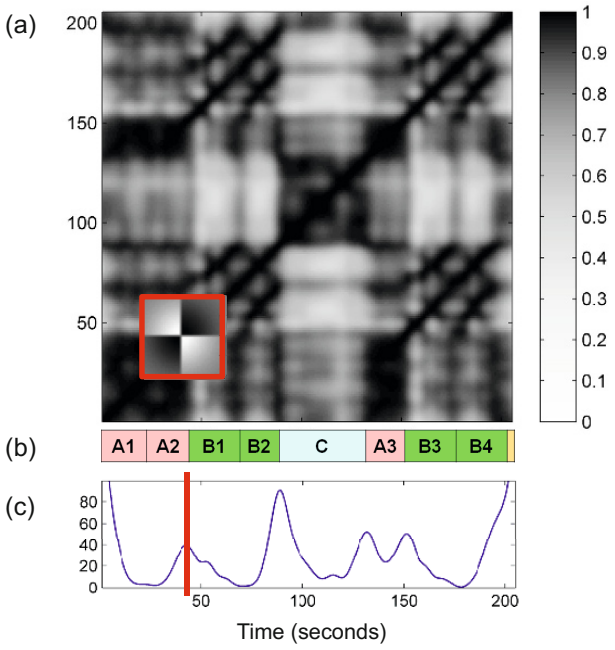


Fig. 4.24 Novelty function obtained by correlating an SSM with a Gaussian checkerboard kernel for a recording of the Hungarian Dance No. 5 by Johannes Brahms. **(a)** SSM similar to the one of Figure 4.10c. **(b)** Manually generated annotation of the musical structure. **(c)** Novelty function.

$k,\ell \in [-L:L]$ (see Figure 4.23c). Finally, to compensate for the influence of the actual kernel size and of the tapering, one may normalize the kernel. This can be done by dividing the kernel by the sum over the absolute values of the kernel matrix:

$$\mathbf{K}_{\mathrm{Norm}}(k,\ell) = \frac{\mathbf{K}_{\mathrm{Gauss}}(k,\ell)}{\sum_{k,\ell\in[-L:L]} |\mathbf{K}_{\mathrm{Gauss}}(k,\ell)|}. \tag{4.42}$$

The normalization becomes important when combining and fusing novelty information that is obtained from kernels of different size.

Now, to detect 2D corner points between adjoining blocks, the idea is to locally compare the SSM with a suitable checkerboard kernel. To this end, we slide a suitable checkerboard kernel $\mathbf{K}$ along the main diagonal of the SSM and sum up the element-wise product of $\mathbf{K}$ and $\mathbf{S}$:

$$\Delta_{\mathrm{Kernel}}(n) := \sum_{k,\ell\in[-L:L]} \mathbf{K}(k,\ell)\mathbf{S}(n+k,n+\ell) \tag{4.43}$$

for $n \in [L+1:N-L]$. Extending the matrix $\mathbf{S}$ on the boundaries by zero-padding (i.e., by setting $\mathbf{S}(k,\ell) = 0$ for $(k,\ell) \in \mathbb{Z}\times\mathbb{Z} \setminus [1:N]\times[1:N]$), one may assume $n \in [1:N]$. This defines a function $\Delta_{\mathrm{Kernel}} : [1:N] \to \mathbb{R}$, also referred to as the **novelty function**, which specifies for each index $n \in [1:N]$ of the feature sequence a measure of novelty $\Delta_{\mathrm{Kernel}}(n)$. When the kernel $\mathbf{K}$ is positioned within a relatively uniform region of $\mathbf{S}$, the positive and negative values of the product tend to sum to zero and $\Delta_{\mathrm{Kernel}}(n)$ becomes small. Conversely, when the kernel $\mathbf{K}$ is positioned exactly at the crux of a checkerboard-like structure of $\mathbf{S}$, the values of the product are all positive and sum up to a large value $\Delta_{\mathrm{Kernel}}(n)$. Figure 4.24c shows a novelty function for our Brahms example using a chroma-based self-similarity matrix. The local maxima of the novelty function nicely indicate changes of harmony, which

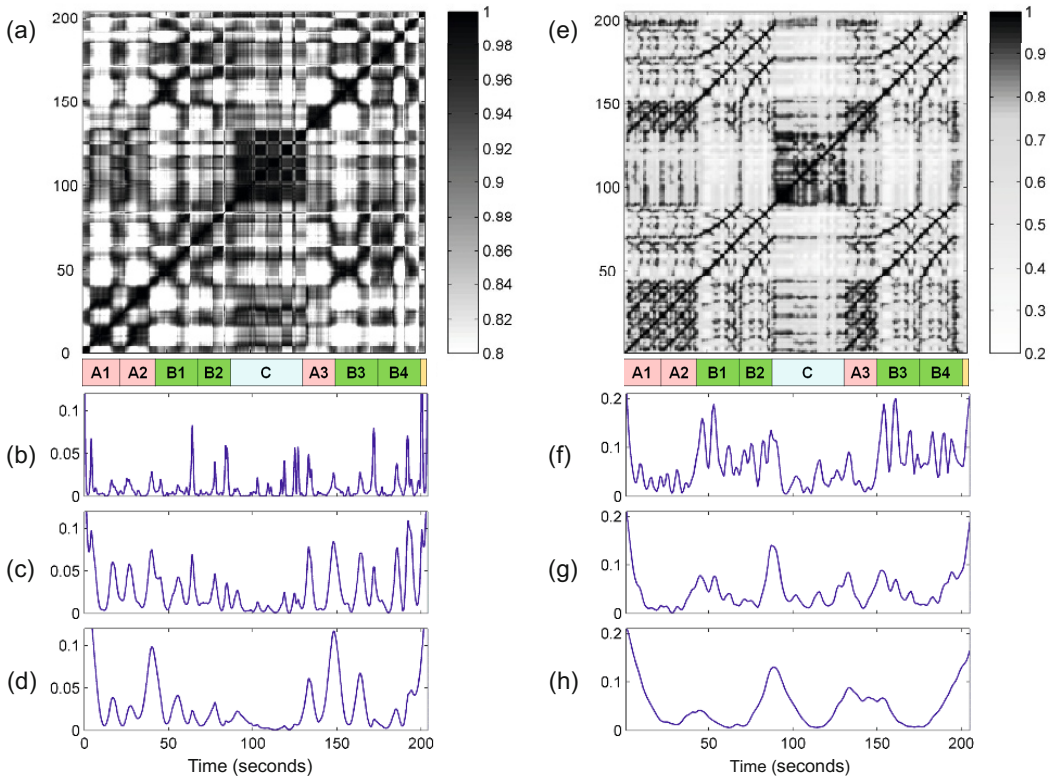


Fig. 4.25 Dependency of novelty functions on the feature representation and the kernel size. **(a)** SSM from Figure 4.7d using tempo-based features. **(b–d)** Novelty functions derived from (a) using a kernel of small/medium/large size. **(e)** SSM from Figure 4.7b using chroma-based features. **(f–h)** Novelty functions derived from (e) using a kernel of small/medium/large size.

particularly occur at boundaries between segments corresponding to different musical parts.

The size of the kernel has a significant impact on the properties of the novelty function. A small kernel may be suitable for detecting novelty on a short time scale, whereas a large kernel is suited for detecting boundaries and transitions between coarse structural sections. The suitability of a given kernel very much depends on the respective application and also on the properties of the underlying self-similarity matrix. This fact is illustrated by Figure 4.25, which shows novelty functions using different sizes and SSMs based on different features. Using a small kernel size may lead to a rather noisy novelty function with many spurious peaks. This particularly holds when the underlying SSM contains not only blocks but also path-like structures as is the case with the SSM shown in Figure 4.25e. Using a larger kernel averages out local fluctuations and results in a smoother novelty function. Note that a similar effect may be achieved by smoothing the SSM, which often leads to an enhancement of the block and an attenuation of the path structure. This effect becomes evident when comparing Figure 4.24 and Figure 4.25e.

In conclusion, the local maxima or peaks of a novelty function correspond to changes in the audio recording. These points often serve as good candidates for boundaries of neighboring segments that correspond to contrasting musical parts. In practice, there are many ways for computing novelty functions and for finding the

relevant peaks. Besides the size of the kernel, a novelty function crucially depends on the characteristics of the underlying self-similarity matrix. In particular, the proposed novelty detection approach is only meaningful when the SSM has block-like structures, which are important in homogeneity-based structure analysis. Moreover, the peak selection strategy is also a delicate step that may have a substantial influence on the quality of the final result. Often, adaptive thresholding strategies where a peak is only selected when its value exceeds a local average of the novelty function are applied. To further reduce the number of spurious peaks, another strategy is to impose a constraint on the minimal distance between two subsequent peak positions.

In the following section, we describe a different approach for novelty detection, which makes it possible to identify structural changes as occurring in repetition-based structure analysis.

4.4.2 Structure Features

Most approaches for novelty detection are performed on the basis of features that capture local characteristics of the given music signal. For example, MFCC-based or chroma-based features capture local characteristics related to timbre or harmony, respectively. Then, a measure of novelty is computed by applying a local kernel or a type of derivative operator based on such feature representations. Computing local differences based on localized features makes such approaches vulnerable to more or less random noise-like fluctuations. We now describe a novelty detection procedure that incorporates global structural properties that go beyond local musical aspects such as harmony or timbre. To this end, we introduce **structure features** on the basis of which various structure analysis principles can be integrated within a unifying framework. The idea behind structure features is to jointly consider local and global aspects by measuring for each frame of a given feature sequence the relations to all other frames of the same feature sequence. This yields a frame-wise, i.e., **local**, feature representation that captures **global** structural characteristics of a feature sequence. The resulting structure features can then be used in combination with standard novelty detection procedures.

We start by introducing the concept of **time-lag matrices**, which is the main technical ingredient for defining the structure features. Let $\mathbf{S}$ be a self-similarity matrix derived from a feature sequence $X = (x_1, x_2, \ldots, x_N)$. Recall that two repeating segments, say $\alpha_1 = [s_1 : t_1]$ and $\alpha_2 = [s_2 : t_2]$, are revealed by a path of high similarity in $\mathbf{S}$ starting at (s_1, s_2) and ending at (t_1, t_2). Furthermore, if there is no relative tempo difference between the two segments, then the path runs exactly parallel to the main diagonal. One may also express this property by saying that segment α_1 is repeated after some time lag corresponding to $\ell = s_2 - s_1$ frames. This observation leads us to the notion of a time-lag representation of an SSM, where one time axis is replaced by a lag axis. To simplify notation, we assume in the following that the frames are indexed starting with the index $n = 0$. Thus, $X = (x_0, x_1, \ldots, x_{N-1})$

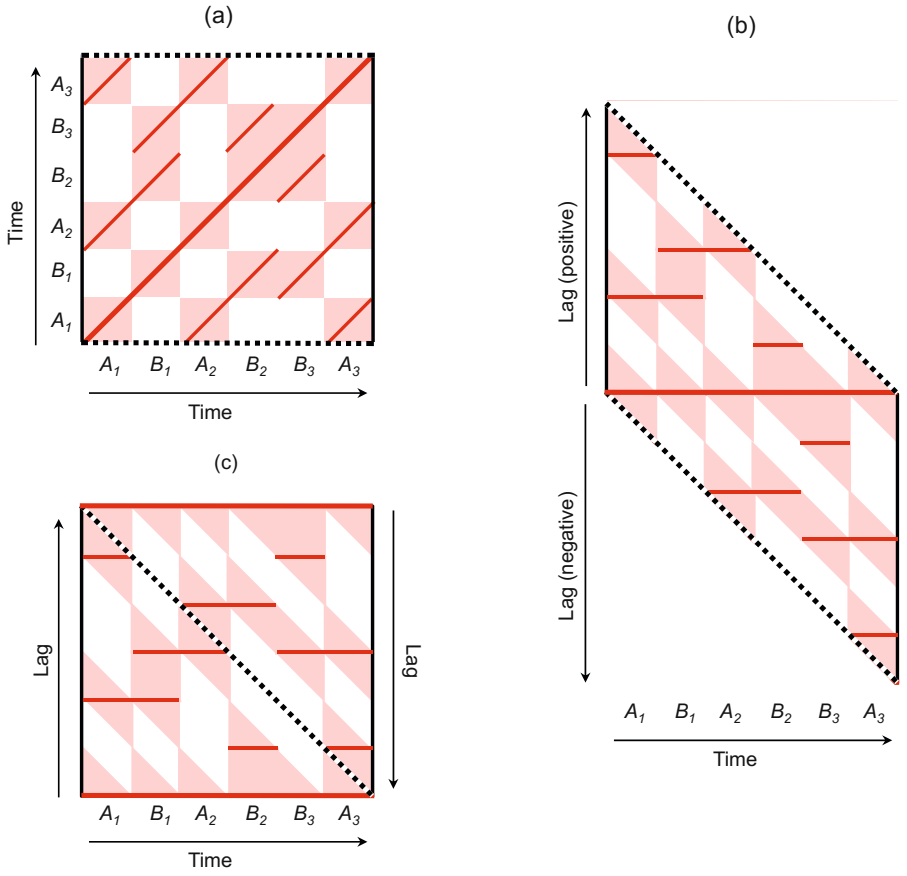


Fig. 4.26 **(a)** Self-similarity matrix **S**. **(b)** Time-lag representation **L**. **(c)** Cyclic time-lag representation of $\mathbf{L}^{\circ}$.

and the self-similarity matrix **S** is indexed by $[0:N-1]\times[0:N-1]$. The **time-lag representation** of **S** is defined by

$$\mathbf{L}(\ell,n)=\mathbf{S}(n+\ell,n) \tag{4.44}$$

for $n\in[0:N-1]$ and $\ell\in[-n:N-1-n]$. Note that the range for the lag parameter ℓ depends on the time parameter n. The lag index must be chosen in such a way that the sum $n+\ell$ lies in the range $[0:N-1]$. For example, for time index $n=0$ one can only look into the future with $\ell\in[0:N-1]$, whereas for time index $n=N-1$ one can only look into the past with $\ell\in[-N+1:0]$. As an example, Figure 4.26a shows a self-similarity matrix **S** and Figure 4.26b its time-lag representation **L**, which is obtained by shearing the original matrix parallel to the horizontal axis. As a result, lines that are parallel to the main diagonal in **S** become horizontal lines in **L**. In other words, diagonal structures are transformed into horizontal structures. To simplify notation, we also introduce the **circular time-lag representation** $\mathbf{L}^{\circ}$ by defining

$$\mathbf{L}^{\circ}(\ell,n)=\mathbf{S}((n+\ell)\bmod N,n) \tag{4.45}$$

for $n \in [0:N-1]$ and $\ell \in [0:N-1]$. As also illustrated by Figure 4.26c, a negative time-lag parameter $\ell \in [-n:-1]$ as used in $\mathbf{L}$ is identified with $\ell + N$ in $\mathbf{L}^{\circlearrowleft}$. Doing so, the time-lag representation $\mathbf{L}^{\circlearrowleft}$ again becomes a matrix indexed by $[0:N-1] \times [0:N-1]$ as for the matrix $\mathbf{S}$.

What have we gained by considering a time-lag representation of a self-similarity matrix? In the following, let $\mathbf{S}^{[n]}$ denote the n^{th} column of $\mathbf{S}$ for a given time frame $n \in [0:N-1]$. Recall that the vector $\mathbf{S}^{[n]} \in \mathbb{R}^N$ reveals the kind of relations that exist for time frame n. In the case that $\mathbf{S}^{[n]}(m)$ is large for some $m \in [0:N-1]$, then time frame n is related to time frame m. In the case that the value is small, the two frames are unrelated. In other words, $\mathbf{S}^{[n]}$ reveals the global structural relations of frame n. The same interpretation holds for the n^{th} column of the time-lag matrix $\mathbf{L}^{\circlearrowleft[n]}$. However, there is a crucial difference between $\mathbf{S}$ and $\mathbf{L}^{\circlearrowleft}$. In the case that two subsequent frames n and $n+1$ have the same structural properties, the two vectors $\mathbf{S}^{[n]}$ and $\mathbf{S}^{[n+1]}$ are **cyclically shifted** versions of each other, whereas the two vectors $\mathbf{L}^{\circlearrowleft[n]}$ and $\mathbf{L}^{\circlearrowleft[n+1]}$ are **identical**.

Based on this observation, we define the **structure features** to be the columns $y_n := \mathbf{L}^{\circlearrowleft[n]} \in \mathbb{R}^N$ for $\mathbf{L}^{\circlearrowleft}$, $n \in [0:N-1]$. By this process, we have converted the original sequence $X = (x_0, x_1, \ldots, x_{N-1})$ of features x_n that capture local (acoustic, musical) characteristics into a sequence $Y = (y_0, y_1, \ldots, y_{N-1})$ of features y_n that capture global (structural) characteristics. As a result, boundaries of the global structural parts can be identified by looking for local changes in the feature sequence Y. There are many ways to capture such local changes. A simple strategy is to compute the difference between successive structure features based on a suitable distance function. For example, using the Euclidean norm of $\mathbb{R}^N$ (see (2.38)), one obtains a novelty function

$$\Delta_{\text{Structure}}(n) := \|y_{n+1} - y_n\| = \|\mathbf{L}^{\circlearrowleft[n+1]} - \mathbf{L}^{\circlearrowleft[n]}\| \tag{4.46}$$

for $n \in [0:N-2]$. Again, by zero-padding one may assume $n \in [0:N-1]$. The positions of local maxima or peaks of this function yield candidates for structural boundaries. The overall procedure depends on many design choices and parameter settings including the feature type used for the original sequence X or the way $\mathbf{S}$ is computed. Also, in practice, one often uses more involved derivative operators and applies suitable preprocessing steps (e.g., further enhancing the matrix $\mathbf{L}^{\circlearrowleft}$) and postprocessing steps (e.g., normalizing the novelty function $\Delta_{\text{Structure}}$). Finally, as already mentioned in Section 4.4.1, the peak selection strategy may have a crucial influence on the final result.

We close this section by considering the example shown in Figure 4.27, which illustrates the overall procedure for structure-based novelty detection. The underlying piece of music is the Mazurka Op. 24, No. 1 by Frédéric Chopin, which has the musical structure $A_1A_2B_1B_2A_3A_4B_3B_4CA_5A_6$. Figure 4.27a shows a path-enhanced and binarized SSM computed from a chroma-based feature representation of an audio recording. The resulting circular time-lag representation $\mathbf{L}^{\circlearrowleft}$ and novelty function $\Delta_{\text{Structure}}$ are shown in Figure 4.27b and Figure 4.27c, respectively. Note that the peak positions of $\Delta_{\text{Structure}}$ coincide well with the (joint) start and end positions of path components, which in turn concur with boundaries of the musical sections.

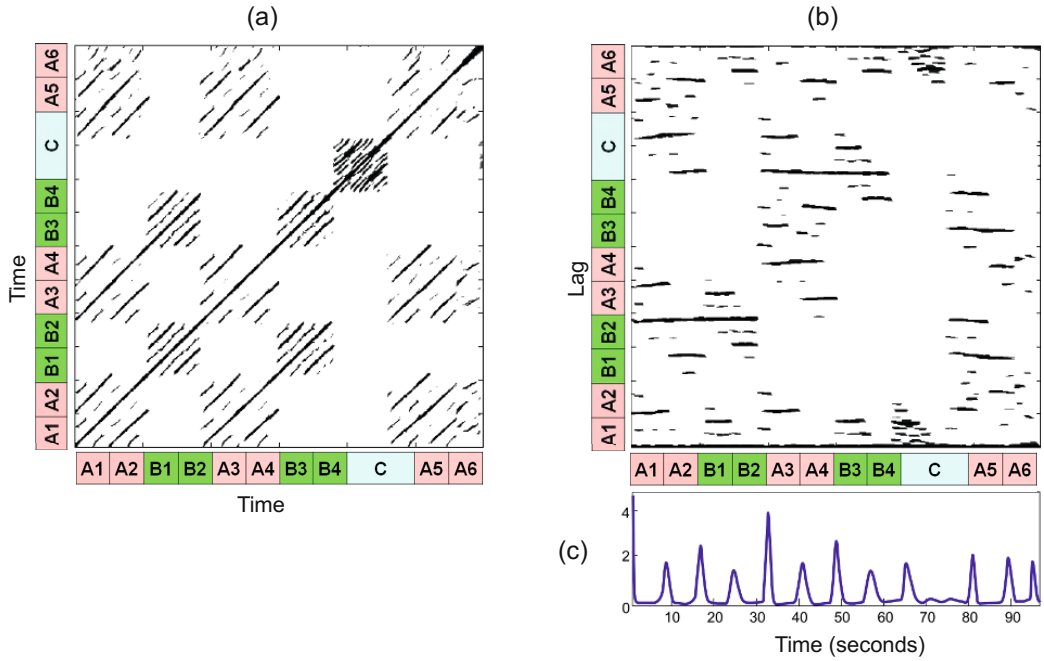


Fig. 4.27 Novelty-based segmentation using structure features for a recording of the Mazurka Op. 24, No. 1 by Frédéric Chopin. **(a)** Path-enhanced and binarized self-similarity matrix **S**. **(b)** Circular time-lag representation $\mathbf{L}^{\circ}$. **(c)** Novelty function $\Delta_{\text{Structure}}$.

The structure features work particularly well in this example because of two reasons. First, there are many repeating parts, resulting in a rich path structure. Second, the various repeating musical parts occur in different chronological orders, resulting in characteristic path discontinuations that are captured well by the structure features; That is, structure-based novelty detection does not work for a piece with musical structure $A_1A_2A_3A_4$ or $A_1B_1A_2B_2$, but works well for a piece with musical structure $A_1B_1A_2A_3$ or $A_1A_2B_1B_2$ (see Exercise 4.14).

4.5 Evaluation

We have described various procedures for extracting structural information from a given music recording. However, we have not yet discussed the issue of measuring **how well** a given procedure performs the task at hand. In this section, we address the problem of automatically evaluating structure analysis algorithms and explain why the evaluation itself constitutes a nontrivial task.

A general evaluation approach in structure analysis is to compare an **estimated result** obtained by some automated procedure against some **reference result**. To realize such a general approach, one needs to find answers to the following questions: How is a structure analysis result actually modeled? How should the estimated result be compared against the reference result? Where does the reference result come from and is it reliable? In particular the last question easily leads to philosophical

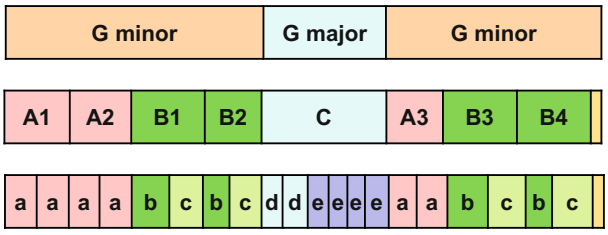


Fig. 4.28 Structure annotation on various scales of the Hungarian Dance No. 5 by Johannes Brahms.

considerations on the nature and meaning of musical structures. As we have already discussed in Section 4.1 and illustrated by Figure 4.2, music structure analysis is an ill-posed problem that depends on many different factors, not to mention the musical and acoustic variations that occur in real-world music recordings. Since a structure analysis result largely depends on the musical context and the considered temporal level, even two human experts may disagree in their analysis of a given piece of music. In the case of our Brahms example, as we have already discussed in Section 4.1.2 and as illustrated by Figure 4.28, one expert may annotate the structure on a larger scale resulting in the musical structure $A_1A_2B_1B_2CA_3B_3B_4D$, while another expert may consider a smaller scale, where the parts are further subdivided.

For the moment, we do not dwell on the latter issue any further. Instead, we assume that a valid reference structure annotation has been provided by a human expert, even though this is a simplistic and sometimes problematic assumption. Such an annotation is also often referred to as **ground truth**. The objective of the automated procedure is to estimate a structure annotation that is as close to the reference as possible. After introducing some general notions (Section 4.5.1), we discuss some evaluation metrics often used for comparing structure analysis results (Section 4.5.2).

4.5.1 Precision, Recall, F-Measure

Many evaluation measures are based on some notion of precision, recall, and F-measure—a concept that has been borrowed from the fields of information retrieval and pattern recognition. We now introduce this general concept in the context of binary classification (see Figure 4.29 for an overview). First, let $\mathcal{I}$ be a finite set of so-called **items**. For this set, one has a reference annotation that is the result of a binary classification. Each item $i \in \mathcal{I}$ is assigned either a label '+' (item is **positive** or **relevant**) or a label '−' (item is **negative** or **not relevant**). Let $\mathcal{I}_+^{\mathrm{Ref}}$ be the set of positive items, and $\mathcal{I}_-^{\mathrm{Ref}}$ be the set of negative items. Furthermore, one has an automated procedure that estimates the annotation for each item. Let $\mathcal{I}_+^{\mathrm{Est}}$ be the set of items being estimated as positive, and $\mathcal{I}_-^{\mathrm{Est}}$ be the set of items being estimated as negative. An item $i \in \mathcal{I}_+^{\mathrm{Est}}$ estimated as positive is called a **true positive** (TP) if it belongs to $\mathcal{I}_+^{\mathrm{Ref}}$, i.e., if $i \in \mathcal{I}_+^{\mathrm{Est}} \cap \mathcal{I}_+^{\mathrm{Ref}}$. Otherwise, if $i \in \mathcal{I}_+^{\mathrm{Est}} \cap \mathcal{I}_-^{\mathrm{Ref}}$, it is called a **false positive** (FP). Similarly, an item $i \in \mathcal{I}_-^{\mathrm{Est}}$ estimated as negative is called a **false negative** (FN) if it belongs to $\mathcal{I}_+^{\mathrm{Ref}}$, and **true negative** (TN) otherwise.

		Reference annotation ("Ground truth")		
		Positive	Negative	
Estimated annotation ("Algorithm")	Positive	True positive (TP)	False positive (FP)	$\mathrm{P} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}}$
	Negative	False negative (FN)	True negative (TN)	
		$\mathrm{R} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}}$		$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$

Fig. 4.29 Definition of precision, recall, and F-measure.

The **precision** P of the estimation is defined as the number of true positives divided by the total number of items estimated as positive:

$$\mathrm{P} = \frac{|\mathcal{I}_+^{\mathrm{Est}} \cap \mathcal{I}_+^{\mathrm{Ref}}|}{|\mathcal{I}_+^{\mathrm{Est}}|} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}}. \tag{4.47}$$

In contrast, the **recall** R is defined as the number of true positives divided by the total number of positive items:

$$\mathrm{R} = \frac{|\mathcal{I}_+^{\mathrm{Est}} \cap \mathcal{I}_+^{\mathrm{Ref}}|}{|\mathcal{I}_+^{\mathrm{Ref}}|} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}}. \tag{4.48}$$

Note that both precision and recall have values in the interval $[0, 1]$. A perfect precision $\mathrm{P} = 1$ means that every item estimated as positive is indeed positive. In this case, there is no false positive, but there may exist some false negatives. In contrast, a perfect recall $\mathrm{R} = 1$ means that every positive item was also estimated as positive. In this case, there is no false negative, but there may exist some false positives. Only in the case $\mathrm{P} = 1$ and $\mathrm{R} = 1$ does the estimated annotation coincide with the reference annotation. Precision and recall are often combined by taking their harmonic mean to form a single measure, often referred to as the **F-measure**:

$$\mathrm{F} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}. \tag{4.49}$$

The harmonic mean is further discussed in Exercise 4.8. One main property is that $\mathrm{F} \in [0, 1]$ with $\mathrm{F} = 1$ if and only if $\mathrm{P} = 1$ and $\mathrm{R} = 1$.

4.5.2 Structure Annotations

With these formal definitions at hand, let us come back to the structure analysis scenario. Since there are many different analysis tasks and aspects to be consid-

ered, it is not at all clear how a mathematical model for the analysis result has to be specified. Let us start with the general task of deriving the musical structure from a given audio recording. In the following, we consider the discrete-time case, where the sampled time axis is indexed by $[1:N]$. We call the result of a structure analysis a **structure annotation**, which consists of a segmentation of the time axis together with a labeling of the segments. The segmentation is modeled by a segment family $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ of some size K as introduced in (4.20). Note that at this stage we make the assumption that segments are disjoint. On the one hand, this is a convenient restriction, which simplifies the comparison of different structure annotations. On the other hand, this assumption may not always be appropriate. For example, it does not allow to capture hierarchical and nested structures. For the labeling, let Λ be a set of possible labels. For example, Λ may be the set $\{A, B, C, \ldots, a, b, c, \ldots\}$, it may consist of a set of suitable strings such as $\{\text{Chorus}, \text{Verse}, \text{Bridge}, \ldots\}$, or it may simply be a subset of $\mathbb{N}$. Then the **labeling** can be modeled by assigning to each segment α_k a label $\lambda_k \in \Lambda$, $k \in [1:K]$. To simplify notation, we additionally assume that the segment family covers the entire time axis (i.e., $\bigcup_{k=1}^{K} \alpha_k = [1:N]$) so that each frame index $[1:N]$ is assigned to exactly one label. In Exercise 4.15, we show that this assumption does not lead to any loss of generality.

There are many ways to compare an estimated structure annotation with a reference annotation and for deriving some kind of "success" measure. In its strictest form, one could simply say that either the two annotations are identical ("success") or not ("fail"). In practice, however, such a binary measure is not very meaningful. Instead, one requires measures that indicate the degree of similarity of two given annotations and that is insensitive towards small differences in the annotations. For example, such differences may be due to small shifts in segment boundaries or local deviations in the labeling. Furthermore, even though two annotations may be based on the same segmentation and the same grouping of segments, they may differ in the naming of the labels. For example, in the reference annotation, segments may be labeled by strings such as "verse" or "chorus" whereas in the estimated annotation corresponding parts may be labeled by letters such as A or B. In many applications, such a mismatch in the label naming is not considered to be a failure of the algorithm. Having these issues in mind, we now discuss some evaluation measures in more detail.

4.5.3 Labeling Evaluation

We start with some purely frame-based evaluation measures, which are referred to as **pairwise** precision, recall, and F-measure. In these measures, the segment boundaries are left unconsidered and only the labeling information is used. For a given structure annotation, we define a **label function** $\varphi : [1:N] \to \Lambda$ by setting

$$\varphi(n) := \lambda_k \tag{4.50}$$

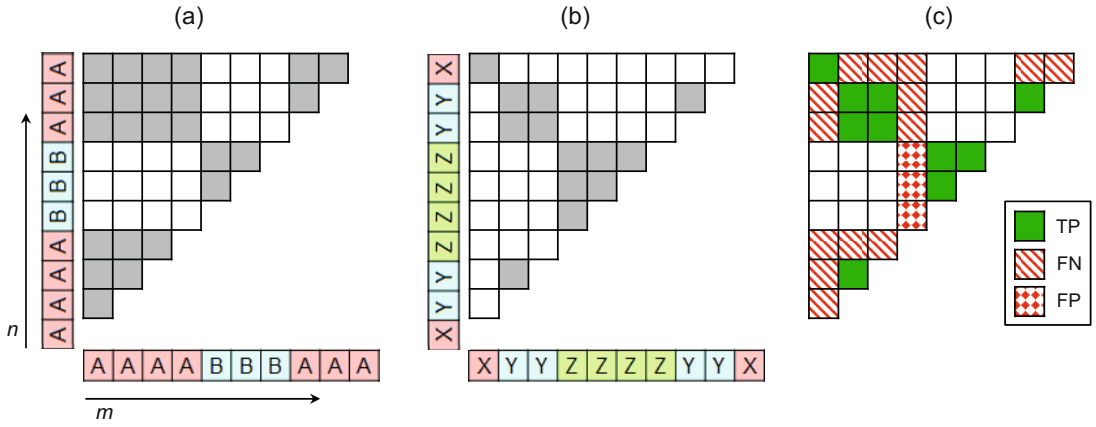


Fig. 4.30 Illustration of pairwise precision, recall, and F-measure. **(a)** Positive items (indicated by gray boxes) with regard to the reference annotation. **(b)** Positive items (indicated by gray boxes) with regard to the estimated annotation. **(c)** True positive (TP), false positive (FP), and false negative (FN) items.

for $n \in \alpha_k$ (assuming that the segment family covers the entire time axis). Let φ^{Ref} and φ^{Est} be the label functions for the reference and estimated structure annotation, respectively. In order to become independent of the actual label naming, the main idea is to not directly look at the labels, but to look for label co-occurrences. To this end, we consider pairs of frames that are assigned to the same label. More precisely, we define the set

$$\mathcal{I} = \{(n,m) \in [1:N] \times [1:N] \mid m < n\}, \tag{4.51}$$

which serves as a set of items as described in Section 4.5.1. For the reference and estimated annotations, we define the positive items by

$$\mathcal{I}_+^{\mathrm{Ref}} = \{(n,m) \in \mathcal{I} \mid \varphi^{\mathrm{Ref}}(n) = \varphi^{\mathrm{Ref}}(m)\}, \tag{4.52}$$

$$\mathcal{I}_+^{\mathrm{Est}} = \{(n,m) \in \mathcal{I} \mid \varphi^{\mathrm{Est}}(n) = \varphi^{\mathrm{Est}}(m)\}, \tag{4.53}$$

whereas $\mathcal{I}_-^{\mathrm{Ref}} = \mathcal{I} \setminus \mathcal{I}_+^{\mathrm{Ref}}$ and $\mathcal{I}_-^{\mathrm{Est}} = \mathcal{I} \setminus \mathcal{I}_+^{\mathrm{Est}}$. In other words, an item (n,m) is considered to be positive with regard to an annotation if the frames n and m have the same label. Now, the **pairwise precision** is defined to be the precision of this binary classification scheme. Similarly, the **pairwise recall** is the recall and the **pairwise F-measure** is the F-measure of this scheme.

The definitions are illustrated by Figure 4.30, where the sampled time interval $[1:N]$ consists of $N = 10$ samples. The reference structure annotation consists of three segments labeled with A, B, and A, respectively. As shown by Figure 4.30a, 24 out of the 45 items are positive with regard to the reference annotation. Similarly, Figure 4.30b shows an estimated structure annotation and the resulting 13 positive items. In Figure 4.30c, the true positives (#TP = 10), false positives (#FP = 3), and false negatives (#FN = 14) are indicated. From this, one obtains

$$\mathrm{P} = \#\mathrm{TP}/(\#\mathrm{TP}+\#\mathrm{FP}) = 10/13 \approx 0.769, \tag{4.54}$$

$$\mathrm{R} = \#\mathrm{TP}/(\#\mathrm{TP}+\#\mathrm{FN}) = 10/24 \approx 0.417, \tag{4.55}$$

$$\mathrm{F} = 2\mathrm{PR}/(\mathrm{P}+\mathrm{R}) \approx 0.541. \tag{4.56}$$

In this example, the precision of nearly 77% is relatively high, whereas the recall of 42% is relatively low. The F-measure is between these two values with a bias towards the smaller one. Further examples are discussed in the exercises.

4.5.4 Boundary Evaluation

The pairwise precision, recall, and F-measure are solely based on label information, whereas segment boundaries are treated implicitly by the presence of label changes. For other structure analysis tasks such as novelty-based segmentation, the precise detection of boundaries is the focus. To evaluate such procedures, one measures the deviation of the estimated segment boundaries from the boundaries of a reference annotation. To mathematically model this scenario, we introduce the notion of a **boundary annotation**, which is given by a sequence $B = (b_1, b_2, \ldots, b_K)$ of increasing indices $b_k \in [1:N]$, $k \in [1:K]$. For example, such a boundary annotation may be derived from a structure annotation by taking the start and possibly the end indices of the annotated segments. In the following, let B^{Ref} be the reference boundary annotation and B^{Est} the estimated boundary annotation. There are many ways to compare B^{Est} against B^{Ref}. For example, using $\mathcal{I} = [1:N]$ as a set of items, one can define $\mathcal{I}_+^{\mathrm{Ref}} := B^{\mathrm{Ref}}$ and $\mathcal{I}_+^{\mathrm{Est}} := B^{\mathrm{Est}}$. From this, the precision, recall, and F-measure can be computed in the usual way. In this case, an estimated boundary is considered correct only if it agrees with a reference boundary.

For certain applications small deviations in the boundary positions are acceptable. Therefore, one generalizes the previous measures by introducing a tolerance parameter $\tau \geq 0$ for the maximal acceptable deviation. An estimated boundary $b^{\mathrm{Est}} \in B^{\mathrm{Est}}$ is then considered **correct** if it lies within the τ-neighborhood of a reference boundary $b^{\mathrm{Ref}} \in B^{\mathrm{Ref}}$:

$$|b^{\mathrm{Est}} - b^{\mathrm{Ref}}| \leq \tau. \tag{4.57}$$

In this case, the sets $\mathcal{I}_+^{\mathrm{Ref}}$ and $\mathcal{I}_+^{\mathrm{Est}}$ can no longer be used for defining precision and recall. Instead, we generalize the notions of true positives, false positives, and false negatives. The **true positives** (TP) are defined to be the items $b^{\mathrm{Est}} \in B^{\mathrm{Est}}$ that are correct, and the **false positives** (FP) are the items $b^{\mathrm{Est}} \in B^{\mathrm{Est}}$ that are not correct. Furthermore, the **false negatives** (FN) are defined to be the items $b^{\mathrm{Ref}} \in B^{\mathrm{Ref}}$ with no estimated item in a τ-neighborhood. Based on these definitions, one can compute precision, recall, and F-measure from #TP, #FP, and #FN using the formulas of Figure 4.29.

However, this generalization needs to be taken with care. Because of the tolerance parameter τ, several estimated boundaries may be contained in the τ-neighborhood of a single reference boundary. Conversely, a single estimated bound-

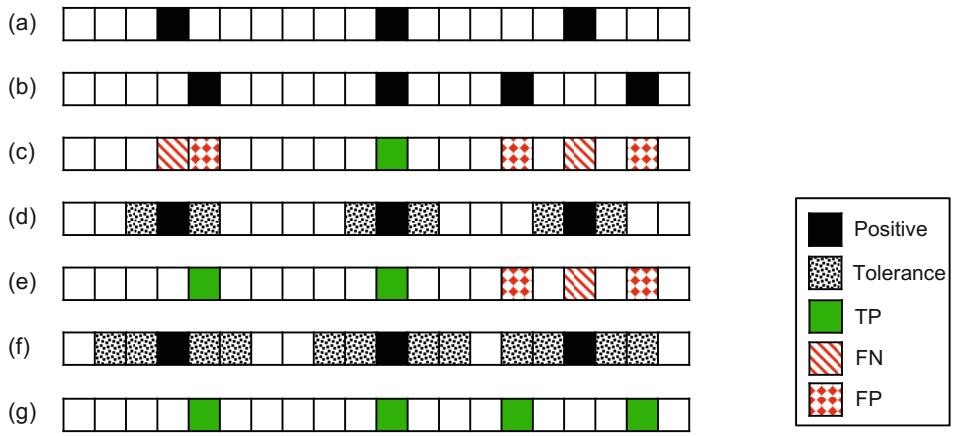


Fig. 4.31 Illustration of boundary evaluation. **(a)** Reference boundary annotation. **(b)** Estimated boundary annotation. **(c)** Evaluation of (b) with regard to (a). **(d)** τ-Neighborhood of (a) using the tolerance parameter $\tau = 1$. **(e)** Evaluation of (b) with regard to (d). **(f)** τ-Neighborhood of (a) using the tolerance parameter $\tau = 2$. **(g)** Evaluation of (b) with regard to (f).

ary may be contained in the τ-neighborhood of several reference boundaries. As a result, one may obtain a perfect F-measure even in the case that the sets B^{Est} and B^{Ref} contain a different number of boundaries. From a semantic point of view, this is not meaningful. To avoid such anomalies, one may introduce an additional assumption in the definition of a boundary annotation by requiring

$$|b_{k+1} - b_k| > 2\tau \tag{4.58}$$

for $k \in [1 : N-1]$. This is also a meaningful requirement from a musical point of view: a musical section (determined by two subsequent boundaries) should be much longer than the size of the tolerance parameter.

Figure 4.31 illustrates the boundary evaluation measures by means of a simple example. Using the tolerance parameter $\tau = 0$, one obtains #TP $= 1$, #FP $= 3$, and #FN $= 2$ (see Figure 4.31c). This yields P $= 1/4$, R $= 1/3$, and F $= 2/7$. In the case $\tau = 1$, one obtains #TP $= 2$, #FP $= 2$, and #FN $= 1$, which results in P $= 1/2$, R $= 2/3$, and F $= 4/7$ (see Figure 4.31e). Finally, when using $\tau = 2$, one obtains a perfect F-measure. However, in this case the condition (4.58) is violated and the meaning of the evaluation measure is questionable.

4.5.5 Thumbnail Evaluation

As a third scenario, we now discuss some evaluation measures for audio thumbnailing, which is a prominent subtask of general music structure analysis. In Section 4.3, we introduced an automated procedure for identifying the most representative section from a given audio recording. Mathematically, this section and its repetitions

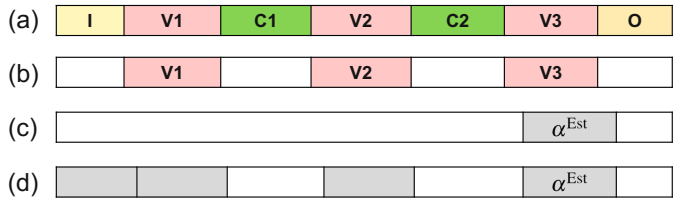


Fig. 4.32 Illustration of thumbnail evaluation. **(a)** Reference structure annotation. **(b)** Reference thumbnail family. **(c)** Estimated thumbnail. **(d)** Segment family of estimated thumbnail.

are modeled by a segment family $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ of some size K (see (4.20)). Each of the segments of this family may serve equally well as the thumbnail.

In the thumbnailing scenario, one does not need an entire structure annotation, but only the label and the associated segment family that represents the thumbnail and its repetitions. For example, for a popular song, the thumbnail may be the verse part of the song so that the associated segment family would consist of all verse sections of the recording (see Figure 4.32b). As in the previous evaluation scenarios, we assume that a suitable reference annotation is available. This annotation is given in the form of a segment family of the audio thumbnail, which is denoted by $\mathcal{A}^{\mathrm{Ref}}$ and also referred to as the **reference thumbnail family**. Furthermore, let α^{Est} be the estimated thumbnail segment. Since every segment $\mathcal{A}^{\mathrm{Ref}}$ can serve equally well as the reference thumbnail, we consider the estimated thumbnail α^{Est} to be correct if it agrees (at least to a large degree) with one of these segments. Therefore, to measure how well the estimated thumbnail α^{Est} corresponds to the reference thumbnail family, we compute

$$\mathrm{P}^{\alpha} = \frac{|\alpha^{\mathrm{Est}} \cap \alpha|}{|\alpha^{\mathrm{Est}}|}, \tag{4.59}$$

$$\mathrm{R}^{\alpha} = \frac{|\alpha^{\mathrm{Est}} \cap \alpha|}{|\alpha|}, \tag{4.60}$$

$$\mathrm{F}^{\alpha} = \frac{2\mathrm{P}^{\alpha}\mathrm{R}^{\alpha}}{\mathrm{P}^{\alpha} + \mathrm{R}^{\alpha}} \tag{4.61}$$

for each $\alpha \in \mathcal{A}^{\mathrm{Ref}}$ and define the **thumbnail F-measure** by

$$\mathrm{F}^{\mathrm{Thumb}} = \max\left\{\mathrm{F}^{\alpha} \mid \alpha \in \mathcal{A}^{\mathrm{Ref}}\right\}. \tag{4.62}$$

In other words, the thumbnail F-measure expresses to what extent α^{Est} maximally agrees with one of the reference thumbnails contained in $\mathcal{A}^{\mathrm{Ref}}$.

As an example, Figure 4.32 shows a song having the musical structure $IV_1C_1V_2C_2V_3O$, where the verse part is considered to be the thumbnail. Hence, the reference thumbnail family consists of the three segments corresponding to V_1, V_2, and V_3 (see Figure 4.32b). Figure 4.32c shows the estimated thumbnail segment α^{Est}. Since α^{Est} has no overlap with the V_1-part and V_2-part segments, the corresponding F-measures are zero. However, the F-measure between α^{Est} and the V_3-part segment α is $\mathrm{F}^{\alpha} = 0.8$, thus $\mathrm{F}^{\mathrm{Thumb}} = 0.8$ follows. Even though not needed for the evaluation, Figure 4.32d shows the segment family of the estimated thumb-

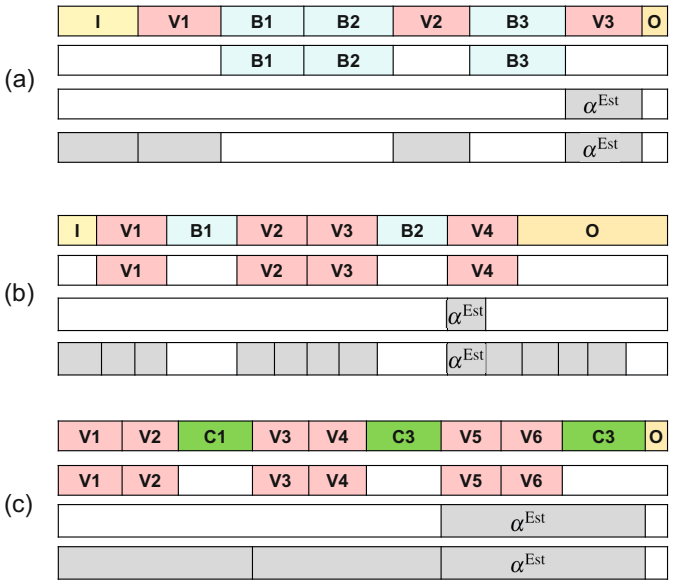


Fig. 4.33 Illustration of typical error sources in thumbnailing and music structure analysis (see Figure 4.32 for an explanation of the annotations). **(a)** Confusion problem for Beatles song "Martha My Dear." **(b)** Substructure (oversegmentation) problem for Beatles song "While My Guitar Gently Weeps." **(c)** Superordinate structure (undersegmentation) problem for Beatles song "For No One."

nail segment α^{Est}, which is obtained when using the fitness-based thumbnailing approach from Section 4.3. This family, which reveals all estimated repetitions of α^{Est}, contains four segments. In this example, it turns out that the intro (the section labeled as I) is harmonically very similar to the three verse sections (labeled as V_1, V_2, and V_3).

We close this section by discussing some typical error sources in audio thumbnailing. As a first example, let us consider the Beatles song "Martha My Dear" shown in Figure 4.33a. The annotated bridge segments were chosen as reference thumbnail family, whereas the estimated thumbnail corresponds to a verse segment (which is actually quite similar to the intro). For this song, the V-part and B-part segments both appear three times and have roughly the same duration. As a result, it is hard to decide whether to use the verse or the bridge for defining the reference segment family.

A second problem occurs when the thumbnail has a substructure. For example, in the Beatles song "While My Guitar Gently Weeps," the verse has a substructure basically consisting of two repeating subparts (see Figure 4.33b). Therefore, a segment that corresponds to the first or second half of the V-part may also serve as a meaningful thumbnail. Such a segment was chosen by the automated procedure as the estimated thumbnail. This is a typical example of a problem generally referred to as **oversegmentation**, where meaningful annotations exist on various scales. We have encountered this phenomenon already in our Brahms example shown in Figure 4.28.

Finally, as illustrated by the Beatles song "For No One" in Figure 4.33c, superordinate repeating parts may also have a high fitness, thus being selected as estimated thumbnails. In this example, the automated procedure identified the superordinate structure VVC (consisting of two verses and a chorus section) as thumbnail. This problem is generally referred to as **undersegmentation**, which is the counterpart to oversegmentation.

These three examples are typical for the kind of problems one has to face when dealing with ill-posed tasks such as music structure analysis. An automated proce-

dure may yield a result that does not coincide with a reference annotation, but is still meaningful from a musical point of view. In such cases, it is less that the procedure has failed and more that the problem is ambiguous.

4.6 Further Notes

In this chapter, we have studied various related research problems commonly subsumed under the name of music structure analysis. The general objective is to segment an audio recording with regard to various musical aspects, for example, identifying recurrent themes or detecting temporal boundaries between contrasting musical parts. The challenge of structure analysis is that music is highly complex and rich. Being organized in a hierarchical way, structure in music arises from various relationships between its basic constituent elements. The principles used to create such relationships include repetition, contrast, variation, and homogeneity. As a consequence, many different approaches to derive musical structures have been developed (see [8, 53] for an overview).

Following [53], we have distinguished between three different classes of methods. First, we have looked at repetition-based methods, which are used to identify recurring patterns. As an important application, we applied such methods in Section 4.3 for audio thumbnailing. Second, we have discussed novelty-based methods, which are used to detect transitions and points of novelty. In Section 4.4, we have studied two such approaches for finding structural boundaries between musical parts. Third, we have considered homogeneity-based methods, which are used to determine passages that are consistent with respect to some musical property. In all three cases, one has to account for different musical dimensions such as melody, harmony, rhythm, or timbre (see Section 4.1.3).

In particular, the importance of repetitions in music has been emphasized in the literature. The principle of repetition plays an important role in virtually any sort of music one can think of [39] and constitutes the basis of music as an art form [59]. Repetition is closely related to notions of coherence, intelligibility, and enjoyment in its perception [49], and studies show that for a large variety of music more than 90% of all musical passages longer than a few seconds in duration are repeated in some way or another at some point in the work [25].

In this chapter, we have only scratched the surface of the kind of structures that exist in music. Within the complex hierarchy of musical structures, we have focused on the level typically referred to as musical structure, which describes the overall layout of a composition. One main challenge in structure analysis is that the notion of repetition can be quite ambiguous. What we refer to as repeating musical sections may include significant variations in the musical content. The principle of variation, where motifs and parts are picked up again in a modified or transformed form [31], is another central aspect of music, which has not been covered in this chapter. Furthermore, we have focused on Western music, which follows different rules and possesses different structures than music from other cultures. Despite a

bias towards Western music, we have introduced some basic notions and techniques that are useful for the structural analysis of other types of music and even other types of time-dependent multimedia data.

4.6.1 Self-Similarity Matrices

As one main tool, we have introduced and discussed the concept of self-similarity matrices. These matrices are of great importance for the analysis not only of music signals but also of general time series. For example, such matrices have been employed under the name "recurrence plot" for the analysis of dynamical systems [13]. Later, Foote [14] introduced self-similarity matrices to the music domain in order to visualize the time structure of audio recordings. Since then, similarity matrices and their relatives have been widely used for various music analysis and retrieval tasks beyond music structure analysis [8, 53]. We have already encountered the related concept of cost matrices in the context of music synchronization in Chapter 3. Such matrices will also play an important role in Chapter 7 in the context of content-based music retrieval and version identification [24, 61].

The first step for computing an SSM is to convert the given audio recording into a suitable feature representation. As also detailed in [53], the properties of the resulting SSM crucially depend on the respective feature type. As examples, we have considered MFCC-based features [10] that correlate to the aspect of timbre [1, 67]. Other features referred to as tempogram [4, 23], rhythmogram [26], or beat spectrogram [16] are used to capture beat, tempo, and rhythmic information—a topic that will be addressed in Chapter 6. In particular, we have considered chroma-based audio features, which are particularly suited for analyzing the structure of repeating melodies and harmonies. We have studied these features in detail in Section 3.1.2. Finally, we have shown that the feature rate has a crucial effect on the final structure analysis result, a fact that has also been emphasized in [40, 52].

One important property of similarity matrices is the appearance of block- and path-like structures of high similarity. In Section 4.2.2, we studied several strategies to enhance such structural properties. To augment path-like structures, most enhancement procedures apply some sort of smoothing filter along the direction of the main diagonal [2, 45, 55, 63]. Such a filtering process is closely related to the concept of **time-delay embedding**, which has been previously used for the analysis of dynamical systems [35]. The multiple filtering approach to deal with relative tempo differences between repeating parts was originally suggested in [45]. Also morphological operations used in image processing to enhance contours and edges [11, 60] can be applied for augmenting path-like structures [21]. Thresholding is another important concept for reducing the noise level in SSMs, which may then simplify the path extraction step [46]. Relative and local thresholding strategies are discussed in [63].

Not only the feature type, but also the window size and the temporal resolution used for feature extraction crucially determine whether blocks or stripes are

formed in an SSM [54, 27, 46, 51]. Block-enhanced SSMs have been used for structure analysis based on matrix factorization [28], a technique that we will encounter in Section 8.3 in a different context. In [22, 38], procedures for converting path structures into block structures are proposed. Such conversions make it possible for algorithms previously designed for homogeneity-based structure analysis to be applied to repetition-based structure analysis. Finally, we have seen that a musical part may be repeated in another key. Using chroma features, Goto [17] has suggested simulating transpositions by cyclic chroma shifts. Based on this idea, transposition-invariant SSMs were originally introduced in [41].

The SM toolbox[1] described in [43] contains MATLAB implementations for computing and enhancing similarity matrices in various ways including the methods described in Section 4.2.2. Besides this specialized toolbox, there are more general toolboxes for processing music and audio data. In particular, we want to mention at this point the comprehensive **MIRtoolbox** provided by Lartillot et al. [29, 30]. This toolbox offers a large number of functions for the extraction of audio features that refer to different musical aspects such as tonality, rhythm, structure, and so on. The MIRtoolbox also supplies a basic function (called **mirsimatrix**) for computing similarity matrices.

4.6.2 Audio Thumbnailing

Finding the repetitive structure of a music recording is a widely studied task within music structure analysis (see, e.g., [2, 7, 9, 19, 34, 40, 46, 48, 52, 55, 56]). Many more references can be found in the overview articles [8, 53]. One application of repetition-based structure analysis is audio thumbnailing, where the objective is to find the most representative and repetitive segment of a given music recording (see [2, 5, 6, 32, 44]). Closely following [44], we have presented in Section 4.3 one of these approaches. To identify repetitions, most approaches extract the path structure from an SSM and apply a clustering step to the pairwise relations obtained from the paths in order to derive entire groups of mutually similar segments. Because of noisy and fragmented path structures due to variations, both steps—path extraction as well as grouping—are error-prone and fragile. In [19], a grouping process is described that balances out inconsistencies in the path relations by exploiting a constant tempo assumption. However, when dealing with varying tempo, the grouping process constitutes a challenging research problem [9, 46]. One main contribution in [44] is to jointly perform the path extraction and grouping steps. This idea is realized by assigning a fitness value to a given segment in such a way that all existing relations within the entire recording are simultaneously accounted for. In other words, instead of extracting individual paths, entire groups of paths (encoded by the concept of path families) are extracted, whereby consistency properties within a group are automatically enforced by the construction.

[1] `http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox`

The general idea of assigning a fitness value to each segment of the audio recording has already been formulated by Cooper and Foote [6]. In this early work, the authors calculate the fitness of a given segment as the normalized sum of the self-similarity between the segment and the entire recording, which can be thought of as some sort of "summary score." The thumbnail is then defined to be the fitness-maximizing segment. Also, a visualization of the fitness over all possible segments has been indicated in [6]. However, as one main limitation, the fitness measure does not take any path relations into account, thus yielding only limited information on the repetitiveness of a segment. In [55], Peeters introduced a fitness measure that is based on a binary-valued diagonal path structure extracted from an SSM. For a given segment, diagonal paths above this segment are considered. The fitness is then defined as the sum of the lengths of these paths, where overlaps between repeating instances are prevented by suitable constraints.

The idea of the fitness measure presented in Section 4.3 and introduced in [44] builds upon and extends the pioneering work described in [6, 55] in various ways. Using a variant of dynamic time warping instead of looking for diagonal paths, the presented approach allows for handling tempo differences between repeating segments. Then, combining a coverage criterion (which is similar to the likelihood in [55]) with a score criterion to define a fitness measure balances out two contradicting principles (large coverage versus high average score). Furthermore, introducing suitable normalization steps, trivial self-explanations similar to [36] are disregarded, so that the resulting fitness measure is well suited to compare repetition properties of segments of different lengths. Finally, there exists an optimization scheme based on dynamic programming to efficiently compute the fitness measure. For visualizing the fitness values in a compact and hierarchical way, we have presented in Section 4.3.2 the concept of scape plots. In the music context, these plots were originally used by Sapp [57, 58] to hierarchically represent harmony in musical scores. In [42], a refinement of the fitness scape plot is described, where the relations between different segments are indicated by some suitable color encoding. The frontispiece of this chapter shows such a refined scape plot representation for our Brahms example.

4.6.3 Segmentation Approaches

In Section 4.4 we addressed the topic of novelty-based segmentation, where the goal was to find boundaries between subsequent musical parts. In Section 4.4.1, we presented the kernel-based approach originally described in the classical paper by Foote [15]. There are many other approaches for boundary detection. For example, Tzanetakis and Cook [69] calculate a Mahalanobis distance between successive frames to yield a novelty function. Using an optimization approach, Jensen [26] performs the boundary detection by minimizing the average distance within blocks (defined by neighboring segment boundaries), while keeping the number of seg-

ments small. An entirely different approach to music segmentation is found in [68], where a machine-learning approach that uses training examples is proposed.

The idea of performing novelty detection by means of structure features, as discussed in Section 4.4.2, was originally presented in [62]. For computing these features, one idea is to transform an SSM into a time-lag representation, a concept that has been used in various structure analysis approaches [17, 55]. One important aspect of the approach in [62] is that it integrates different structure analysis principles within a unifying framework: the structure features capture (global) repetition-based information, which is then analyzed using a (local) novelty-based procedure. However, to obtain a full structure annotation, the grouping needs to be done in a separate postprocessing step. The approach by Paulus and Klapuri [50, 52] also combines different segmentation principles by introducing a cost function for structure annotations that considers block-like as well as path-like structures. The final structure annotation is obtained by minimizing the cost function over all possible annotations. However, this approach requires solving a combinatorial optimization task that is computationally prohibitive. To make the computations feasible, the number of candidate annotations is reduced drastically by applying a novelty-based boundary detection procedure in a separate preprocessing step. So far, most procedures for structure analysis rely only on a single principle or apply different principles in separate steps. The development of unifying yet computationally feasible optimization frameworks, which integrate different principles to yield robust segmentation results, remains a challenging area of research. In this context, the path-to-block conversion procedure described in [22] may open up novel ways for simultaneously applying homogeneity-based and repetition-based structure analysis approaches.

4.6.4 Evaluation and Sources

In Section 4.5 we addressed the topic of evaluating automated structure analysis procedures. Many evaluation measures involve some sort of precision and recall rate. As typical examples, we have seen such metrics for three different scenarios. An overview of further evaluation measures can be found in [33]. Systematic evaluations of different structure analysis methods have been performed over recent years within the Music Information Retrieval Evaluation eXchange (MIREX)[2] campaign, which provides a framework for evaluating various kinds of music processing algorithms. The evaluation tasks are typically defined by the research community under the coordination of the International Music Information Retrieval Systems Evaluation Laboratory at the University of Illinois at Urbana-Champaign [12]. On the MIREX websites, one finds not only the evaluation results of numerous procedures, but also links to information on datasets, annotations, evaluation measures, and relevant literature. One popular dataset with publicly available structure annotations [37] consists of recordings of the twelve studio albums by The Bea-

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

tles. Another dataset consists of the 100 recordings of the RWC Popular Music Database [20] with structure annotations described in [18].

Even though music is highly structured and obeys some general rules, what is interesting about any individual piece of music tends to be the way in which it expands or breaks these rules. As a consequence, evaluating the performance of automated procedures is not as easy as one may think. Even so-called reference annotations made by different human experts may differ significantly [47, 52, 64]. Therefore, to better reflect the ambiguity and richness of musical structures, the evaluation should be based on several annotations, which have been generated by several human experts and are provided on different temporal scales [64]. Then, an automated procedure could be treated just as another "expert" and the estimated results could be compared against the entire pool of different reference annotations (instead of using only a single reference annotation). A given automated procedure could then be considered to work correctly if it produces results that are just about as variable as the human annotations. Only if the results are truly quite different could the procedure be considered to work incorrectly. Rather than automatically extracting a structure annotation from scratch, another interesting research direction is to develop automated procedures that somehow explain an existing annotation. A first procedure in this direction is described in [65], where the relevance of various features is determined in relation to a given annotation.

References

1. J.-J. Aucouturier and F. Pachet, *Improving timbre similarity: How high's the sky*, Journal of Negative Results in Speech and Audio Sciences, 1 (2004).
2. M. A. Bartsch and G. H. Wakefield, *Audio thumbnailing of popular music using chroma-based representations*, IEEE Transactions on Multimedia, 7 (2005), pp. 96–104.
3. M. A. Casey and M. Slaney, *The importance of sequences in musical similarity*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, 2006.
4. A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, *On tempo tracking: Tempogram representation and Kalman filtering*, Journal of New Music Research, 28 (2001), pp. 259–273.
5. W. Chai and B. Vercoe, *Music thumbnailing via structural analysis*, in Proceedings of the ACM International Conference on Multimedia, Berkeley, California, USA, 2003, pp. 223–226.
6. M. Cooper and J. Foote, *Automatic music summarization via similarity analysis*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 2002, pp. 81–85.
7. ———, *Summarizing popular music via structural similarity analysis*, in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 2003, pp. 127–130.
8. R. B. Dannenberg and M. Goto, *Music structure analysis from acoustic signals*, in Handbook of Signal Processing in Acoustics, D. Havelock, S. Kuwano, and M. Vorländer, eds., vol. 1, Springer, New York, NY, USA, 2008, pp. 305–331.
9. R. B. Dannenberg and N. Hu, *Pattern discovery techniques for music audio*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 2002, pp. 63–70.
10. S. B. Davis and P. Mermelstein, *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*, Readings in Speech Recognition, (1990), pp. 65–74.
11. E. R. Dougherty, *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press, Bellingham, WA, USA, 1992.
12. J. S. Downie, *The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research*, Acoustical Science and Technology, 29 (2008), pp. 247–255.
13. J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, *Recurrence plots of dynamical systems*, Europhysics Letters, 4 (1987), pp. 973–977.
14. J. Foote, *Visualizing music and audio using self-similarity*, in Proceedings of the ACM International Conference on Multimedia, Orlando, Florida, USA, 1999, pp. 77–80.
15. ———, *Automatic audio segmentation using a measure of audio novelty*, in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), New York, NY, USA, 2000, pp. 452–455.
16. J. Foote and S. Uchihashi, *The beat spectrum: A new approach to rhythm analysis*, in Proceedings of the International Conference on Multimedia and Expo (ICME), Los Alamitos, California, USA, 2001.
17. M. Goto, *A chorus-section detecting method for musical audio signals*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Hong Kong, China, 2003, pp. 437–440.
18. ———, *AIST annotation for the RWC music database*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2006, pp. 359–360.
19. ———, *A chorus section detection method for musical audio signals and its application to a music listening station*, IEEE Transactions on Audio, Speech, and Language Processing, 14 (2006), pp. 1783–1794.
20. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, *RWC music database: Popular, classical and jazz music databases*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 2002.

21. H. GROHGANZ, *Algorithmen zur strukturellen Analyse von Musikaufnahmen*, PhD thesis, University of Bonn, 2015.
22. H. GROHGANZ, M. CLAUSEN, N. JIANG, AND M. MÜLLER, *Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil, 2013, pp. 209–214.
23. P. GROSCHE, M. MÜLLER, AND F. KURTH, *Cyclic tempogram – a mid-level tempo representation for music signals*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, Texas, USA, 2010, pp. 5522 – 5525.
24. P. GROSCHE, M. MÜLLER, AND J. SERRÀ, *Audio content-based music retrieval*, in Multimodal Music Processing, M. Müller, M. Goto, and M. Schedl, eds., vol. 3 of Dagstuhl Follow-Ups, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012, pp. 157–174.
25. D. B. HURON, *Sweet anticipation: Music and the psychology of expectation*, The MIT Press, 2006.
26. K. JENSEN, *Multiple scale music segmentation using rhythm, timbre, and harmony*, EURASIP Journal on Advances in Signal Processing, (2007).
27. F. KAISER, M. G. ARVANITIDOU, AND T. SIKORA, *Audio similarity matrices enhancement in an image processing framework*, in International Workshop on Content-Based Multimedia Indexing (CBMI), Madrid, Spain, 2011.
28. F. KAISER AND T. SIKORA, *Music structure discovery in popular music using non-negative matrix factorization*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Utrecht, The Netherlands, 2010, pp. 429–434.
29. O. LARTILLOT, *MIRtoolbox 1.5, User's Manual*. `https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/MIRtoolbox1.5Guide/`, Retrieved 10.09.2013, 2013.
30. O. LARTILLOT AND P. TOIVIAINEN, *MIR in Matlab (II): A toolbox for musical feature extraction from audio*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Vienna, Austria, 2007, pp. 127–130.
31. F. LERDAHL AND R. JACKENDOFF, *A Generative Theory of Tonal Music*, MIT Press, 1983.
32. M. LEVY, M. SANDLER, AND M. A. CASEY, *Extraction of high-level musical structure from audio data and its application to thumbnail generation*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, 2006, pp. 13–16.
33. H. LUKASHEVICH, *Towards quantitative measures of evaluating song segmentation*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Philadelphia, USA, 2008, pp. 375–380.
34. N. C. MADDAGE, *Automatic structure detection for popular music*, IEEE Multimedia, 13 (2006), pp. 65–77.
35. N. MARWAN, M. C. ROMANO, M. THIEL, AND J. KURTHS, *Recurrence plots for the analysis of complex systems*, Physics Reports, 438 (2007), pp. 237–329.
36. M. MAUCH, *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*, PhD thesis, Queen Mary University of London, 2010.
37. M. MAUCH, C. CANNAM, M. E. DAVIES, S. DIXON, C. HARTE, S. KOLOZALI, D. TIDHAR, AND M. SANDLER, *OMRAS2 metadata project 2009*, in Late Breaking Demo of the International Conference on Music Information Retrieval (ISMIR), Kobe, Japan, 2009.
38. B. MCFEE AND D. ELLIS, *Analyzing song structure with spectral clustering*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 2014, pp. 405–410.
39. R. MIDDLETON, *Form*, in Key terms in popular music and culture, B. Horner and T. Swiss, eds., Wiley-Blackwell, 1999, pp. 141–155.
40. M. MÜLLER, *Information Retrieval for Music and Motion*, Springer Verlag, 2007.
41. M. MÜLLER AND M. CLAUSEN, *Transposition-invariant self-similarity matrices*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Vienna, Austria, 2007, pp. 47–50.

42. M. Müller and N. Jiang, *A scape plot representation for visualizing repetitive structures of music recordings*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 2012, pp. 97–102.
43. M. Müller, N. Jiang, and H. Grohganz, *SM Toolbox: MATLAB implementations for computing and enhancing similiarty matrices*, in Proceedings of the AES Conference on Semantic Audio, London, UK, 2014.
44. M. Müller, N. Jiang, and P. Grosche, *A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing*, IEEE Transactions on Audio, Speech, and Language Processing, 21 (2013), pp. 531–543.
45. M. Müller and F. Kurth, *Enhancing similarity matrices for music audio analysis*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, 2006, pp. 437–440.
46. ———, *Towards structural analysis of audio recordings in the presence of musical variations*, EURASIP Journal on Advances in Signal Processing, 2007 (2007).
47. O. Nieto, M. Farbood, T. Jehan, and J. P. Bello, *Perceptual analysis of the F-measure to evaluate section boundaries in music*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 2014, pp. 265–270.
48. O. Nieto, E. J. Humphrey, and J. P. Bello, *Compressing music recordings into audio summaries*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Curitiba, Brazil, 2013, pp. 313–318.
49. A. Ockelford, *Repetition in music: theoretical and metatheoretical perspectives.*, vol. 13 of Royal Musical Association Monographs, Ashgate Publishing, 2005.
50. J. Paulus and A. P. Klapuri, *Music structure analysis by finding repeated parts*, in Proceedings of the ACM Audio and Music Computing Multimedia Workshop, Santa Barbara, California, USA, 2006, pp. 59–68.
51. ———, *Acoustic features for music piece structure analysis*, in Proceedings of the International Conference on Digital Audio Effects (DAFx), Espoo, Finland, 2008, pp. 309–312.
52. ———, *Music structure analysis using a probabilistic fitness measure and a greedy search algorithm*, IEEE Transactions on Audio, Speech, and Language Processing, 17 (2009), pp. 1159–1170.
53. J. Paulus, M. Müller, and A. P. Klapuri, *Audio-based music structure analysis*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Utrecht, The Netherlands, 2010, pp. 625–636.
54. G. Peeters, *Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach*, in Computer Music Modeling and Retrieval, vol. 2771 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2004, pp. 143–166.
55. G. Peeters, *Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Vienna, Austria, 2007, pp. 35–40.
56. C. Rhodes and M. A. Casey, *Algorithms for determining and labelling approximate hierarchical self-similarity*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Vienna, Austria, 2007, pp. 41–46.
57. C. S. Sapp, *Harmonic visualizations of tonal music*, in Proceedings of the International Computer Music Conference (ICMC), La Habana, Cuba, 2001, pp. 423–430.
58. ———, *Visual hierarchical key analysis*, ACM Computers in Entertainment, 3 (2005), pp. 1–19.
59. H. Schenker, *Der freie Satz*, Universal, Vienna, 1935.
60. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc., Orlando, Florida, USA, 1984.
61. J. Serrà, E. Gómez, P. Herrera, and X. Serra, *Chroma binary similarity and local alignment applied to cover song identification*, IEEE Transactions on Audio, Speech, and Language Processing, 16 (2008), pp. 1138–1151.

62. J. SERRÀ, M. MÜLLER, P. GROSCHE, AND J. L. ARCOS, *Unsupervised detection of music boundaries by time series structure features*, in Proceedings of the AAAI International Conference on Artificial Intelligence, Toronto, Ontario, Canada, 2012.
63. J. SERRÀ, X. SERRA, AND R. G. ANDRZEJAK, *Cross recurrence quantification for cover song identification*, New Journal of Physics, 11 (2009).
64. J. B. L. SMITH, J. A. BURGOYNE, I. FUJINAGA, D. D. ROURE, AND J. S. DOWNIE, *Design and creation of a large-scale database of structural annotations*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Miami, Florida, USA, 2011, pp. 555–560.
65. J. B. L. SMITH AND E. CHEW, *Using quadratic programming to estimate feature relevance in structural analyses of music*, in Proceedings of the ACM International Conference on Multimedia, 2013, pp. 113–122.
66. M. SUNKEL, S. JANSEN, M. WAND, E. EISEMANN, AND H.-P. SEIDEL, *Learning line features in 3D geometry*, Computer Graphics Forum, 30 (2011), pp. 267–276.
67. H. TERASAWA, M. SLANEY, AND J. BERGER, *The thirteen colors of timbre*, in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2005, pp. 323–326.
68. D. TURNBULL, G. LANCKRIET, E. PAMPALK, AND M. GOTO, *A supervised approach for detecting boundaries in music using difference features and boosting*, in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Vienna, Austria, 2007, pp. 51–54.
69. G. TZANETAKIS AND P. COOK, *Multifeature audio segmentation for browsing and annotation*, in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Platz, NY, USA, 1999, pp. 103–106.

Exercises

Exercise 4.1. Let $\mathcal{F} = \mathbb{R}^D$ be the real vector space of dimension $D \in \mathbb{N}$. Typical similarity measures are based on the Euclidean norm (also referred to as the ℓ^2-norm) defined by

$$\|x\|_2 := \left(\sum_{i=1}^{D} |x(i)|^2\right)^{1/2}$$

for a vector $x = (x(1), x(2), \ldots, x(D))^\top$. From this norm, one can derive the similarity measures $s^{a,b} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ for constants $a \in \mathbb{R}$ and $b \in \mathbb{N}$ by setting

$$s^{a,b}(x,y) = a - \|x-y\|_2^b$$

for $x, y \in \mathcal{F}$. In the following, we consider the case $a = 2$ and $b = 2$. Furthermore, assume that x and y are normalized with respect to the ℓ^2-norm. Show that, in this case, the measure $s^{a,b}$ is simply twice the inner product $\langle x|y\rangle$, which measures the cosine of the angle between x and y.

Exercise 4.2. In (4.11), we have introduced a forward smoothing procedure. This procedure results in a fading out of the paths, in particular when using a large length parameter. To avoid this fading out, one idea is to additionally apply the averaging filter in backward direction. The final self-similarity matrix is then obtained by taking the cell-wise maximum over the forward-smoothed and backward-smoothed matrices. Formalize this procedure by giving a mathematical description. Furthermore, show how the backward smoothing can be realized by forward smoothing considering the time-reversed feature sequence.
[**Hint:** To avoid boundary considerations, assume that $\mathbf{S}$ is suitably zero-padded. The effect of the forward–backward smoothing procedure is illustrated by Figure 4.12d. Another example is shown in Figure 4.15c.]

Exercise 4.3. Let $\mathcal{F} = \mathbb{R}^D$ as in Exercise 4.1 and $s : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ be the similarity measure defined by $s(x,y) := |\langle x|y\rangle|$ for $x, y \in \mathcal{F}$ (see (4.3)). Show that the transposition-invariant self-similarity matrix $\mathbf{S}^{\mathrm{TI}}$ (see (4.15)) is symmetric. Is the transposition index matrix $\mathbf{I}$ (see (4.16)) symmetric? Describe the relation between the matrix $\mathbf{I}$ and its transposed matrix $\mathbf{I}^\top$.

Exercise 4.4. For computing the matrix $\mathbf{S}_{L,\Theta}$ in (4.13), a set Θ of relative tempo differences needs to be specified. Assume that $\theta_{\min}$ is a lower bound and $\theta_{\max}$ is an upper bound for the expected relative tempo differences. For a given number $K \in \mathbb{N}$, determine a set

$$\Theta = \{\theta_1 = \theta_{\min}, \theta_2, \ldots, \theta_{K-1}, \theta_K = \theta_{\max}\}$$

consisting of increasing tempo values that are logarithmically spaced. Write a small computer program for computing this set for the parameters $\theta_{\min} = 0.66$, $\theta_{\max} = 1.5$, and $K = 5$, as well as for $\theta_{\min} = 0.5$, $\theta_{\max} = 2$, and $K = 7$.
[**Hint:** Convert the tempo bounds $\theta_{\min}$ and $\theta_{\max}$ into the log domain by applying a logarithm. Then, linearly sample the resulting interval using K samples and apply an exponential function to the samples.]

Exercise 4.5. In this exercise, we look at the various thresholding strategies introduced in Section 4.2.2.4. Given the matrix

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 4 & 3 & 4 & 3 \\ 1 & 1 & 2 & 2 \\ 5 & 6 & 6 & 5 \end{bmatrix},$$

compute the matrices that are obtained by applying the following thresholding operations:

(a) Global thresholding using $\tau = 4$
(b) Global thresholding using $\tau = 4$ as in (a) with subsequent linear scaling of the range $[\tau, \mu]$ to $[0,1]$ using $\mu := \max\{\mathbf{S}(n,m) \mid n,m \in [1:4]\}$
(c) Global thresholding with subsequent linear scaling as in (b) and applying the penalty parameter $\delta = -1$
(d) Relative thresholding using the relative threshold parameter $\rho = 0.5$
(e) Local thresholding in a column- and rowwise fashion using $\rho = 0.5$

Exercise 4.6. Let $X = (x_1, x_2, \ldots, x_N)$ be a sequence and $\alpha = [s:t] \subseteq [1:N]$ a segment of length $M := |\alpha|$. Show that the optimization procedure for computing an optimal path family over α (as described in Section 4.3.1.2) has a complexity of $O(MN)$ regarding the memory requirements as well as the running time.

Exercise 4.7. Let $X = (x_1, \ldots, x_N)$ be a feature sequence and $\mathbf{S}$ the resulting SSM satisfying the normalization properties (4.18) and (4.19). Let $\mathcal{P}^*$ be an optimal path family over a given segment α. Show that $|\alpha| \leq \sigma(\mathcal{P}^*) \leq N$. In particular, this shows that $\sigma(\mathcal{P}^*) = N$ for $\alpha = [1:N]$.

Exercise 4.8. For two given real numbers $a, b \in \mathbb{R}$, the arithmetic mean is defined by $A(a,b) = (a+b)/2$, the geometric mean by $G(a,b) = \sqrt{ab}$, and the harmonic mean by $H(a,b) = 2ab/(a+b)$. Show that $H(a,b) \leq G(a,b) \leq A(a,b)$, i.e., the geometric mean always lies between the harmonic mean and the arithmetic mean. Furthermore, compute $A(a,b)$, $G(a,b)$, and $H(a,b)$ for the numbers $a = 1$ and $b \in \{1,2,3,4\}$.

Exercise 4.9. **(a)** Let us consider a piece of music having the musical structure $A_1B_1B_2A_2A_3$, where we assume that corresponding parts are repeated in exactly the same way. Furthermore, assume that the A-part and B-part segments are completely unrelated to each other and that a B-part segment has exactly twice the length of an A-part segment. Sketch an idealized SSM for this piece (as in Figure 4.18). Furthermore, determine the fitness values of the segments corresponding to A_1 and B_1, respectively.
(b) Next, consider a piece having the musical structure $A_1A_2A_3A_4$, where the four parts are repeated with increasing tempo. Assume that A_1 lasts 20 seconds, A_2 lasts 15 seconds, A_3 lasts 10 seconds, and A_4 lasts 5 seconds. Again sketch an idealized SSM and determine the fitness values of the four segments corresponding to the four parts.

Exercise 4.10. Let $[1:N]$ be a sampled time axis. Show that the number of different segments $\alpha = [s:t]$ with $s,t \in [1:N]$ and $s \leq t$ is $(N+1)N/2$.

Exercise 4.11. Determine the overall computational complexity of calculating the fitness scape plot as introduced in Section 4.3.2 for a feature sequence $X = (x_1, x_2, \ldots, x_N)$ of length N.
[**Hint:** Use Exercise 4.6 and Exercise 4.10.]

Exercise 4.12. Given a triangular representation of all segments within $[1:N]$ as in Figure 4.19b, visually indicate the following sets of segments:

(a) All segments having a minimal length above a given threshold $\theta \geq 0$
(b) All segments that contain a given segment α
(c) All segments that are disjoint to a given segment α
(d) All segments that contain the center $c(\alpha)$ of a given segment α

Exercise 4.13. Sketch the similarity matrix $\mathbf{S}$ and the circular time-lag matrix $\mathbf{L}^\circ$ as in Figure 4.26c for pieces with the following musical structure:

(a) $AB_1B_2B_3$, where all segments have the same length
(b) AB_1B_2, where the A-part and B_1-part segments have the same length and the B_2-part segment has twice the length (played with half the tempo of B_1)

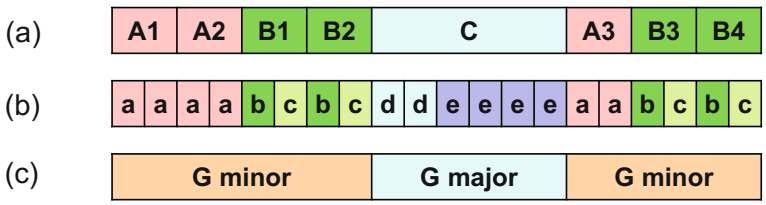
Exercise 4.14. Sketch the similarity matrix $\mathbf{S}$, the circular time-lag matrix $\mathbf{L}^{\circ}$, and the resulting novelty function $\Delta_{\text{Structure}}$ for pieces with the following musical structure (assuming that all segments corresponding to a musical part have the same length and that the kernel size used for computing the novelty function is much smaller than this length):

(a) $A_1A_2A_3A_4$
(b) $A_1B_1A_2B_2$
(c) $A_1B_1A_2A_3$
(d) $A_1A_2B_1B_2$

Exercise 4.15. Let $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ be a segment family together with a labeling $\lambda_k \in \Lambda$, $k \in [1:K]$. Let $\mu(\mathcal{A}) := \bigcup_{k=1}^{K} \alpha_k$ be the union of all segments. Show that one may assume $\mu(\mathcal{A}) = [1:N]$ by suitably extending the segment family, the label set Λ, and the labeling.

Exercise 4.16. In (4.51), we defined the set $\mathcal{I} = \{(n,m) \in [1:N] \times [1:N] \mid n < m\}$ to serve as a set of items for defining the pairwise evaluation measure. Determine the size of $\mathcal{I}$. Furthermore, let $\varphi : [1:N] \to \Lambda$ be a label function, and let $\mathcal{I}_+^{\text{Ref}} = \{(n,m) \in \mathcal{I} \mid \varphi(n) = \varphi(m)\}$ be the set of positive items with regard to φ. Derive a general formula for the size of $\mathcal{I}_+^{\text{Ref}}$.
[**Hint:** Note that the size of $\mathcal{I}_+^{\text{Ref}}$ does not depend on the original order of the frames. Given a specific label, consider the number of frames assigned to that label. To derive a formula for the size of $\mathcal{I}_+^{\text{Ref}}$, one needs to consider all possible labels assumed by φ.]

Exercise 4.17. In this exercise, we investigate how the pairwise labeling evaluation behaves with respect to under- and oversegmentation. To this end, let us consider the following structure annotations of a piece of music (similar to our Brahms example shown in Figure 4.28):



Compute the size $|\mathcal{I}_+|$ for each of the three annotations. Then, assume that (a) is the reference annotation. Compute the pairwise precision, recall, and F-measure for the case that (b) is the estimated annotation ("oversegmentation") and for the case that (c) is the estimated annotation ("undersegmentation").
[**Hint:** Use the results of Exercise 4.16.]

Exercise 4.18. Let $[1:N]$ be a sampled time axis with $N = 50$. Furthermore, let $B^{\text{Ref}} = \{7,13,19,28,40,44\}$ be a reference boundary annotation and $B^{\text{Est}} = \{6,12,21,29,42\}$ be an estimated boundary annotation. Compute the boundary evaluation measures (precision, recall, F-measure) as in Section 4.5.4 for the tolerance parameter $\tau = 0$, $\tau = 1$, and $\tau = 2$, respectively. Why is the case $\tau = 2$ problematic for this example?

Exercise 4.19. Let $[1:N]$ be a sampled time axis with $N = 100$. Furthermore, let $\mathcal{A}^{\text{Ref}} = \{[16:26],[40:49],[50:60],[75:84]\}$ be a reference thumbnail family. Compute the thumbnail F-measure as introduced in Section 4.5.5 for the following estimated thumbnail segments:

(a) $\alpha^{\text{Est}} = [18:27]$
(b) $\alpha^{\text{Est}} = [45:54]$
(c) $\alpha^{\text{Est}} = [60:75]$