Chapter 12
A Wavelet-Based Approach to Pattern Discovery in Melodies

Gissel Velarde, David Meredith, and Tillman Weyde

Abstract We present a computational method for pattern discovery based on the application of the wavelet transform to symbolic representations of melodies or monophonic voices. We model the importance of a discovered pattern in terms of the compression ratio that can be achieved by using it to describe that part of the melody covered by its occurrences. The proposed method resembles that of paradigmatic analysis developed by Ruwet (1966) and Nattiez (1975). In our approach, melodies are represented either as 'raw' 1-dimensional pitch signals or as these signals filtered with the *continuous wavelet transform* (CWT) at a single scale using the Haar wavelet. These representations are segmented using various approaches and the segments are then concatenated based on their similarity. The concatenated segments are compared, clustered and ranked. The method was evaluated on two musicological tasks: discovering themes and sections in the JKU Patterns Development Database and determining the parent compositions of excerpts from J. S. Bach's Two-Part Inventions (BWV 772–786). The results indicate that the new approach performs well at finding noticeable and/or important patterns in melodies and that filtering makes the method robust to melodic variation.

12.1 Introduction

Since the 19th century, music theorists have placed great importance on the analysis of motivic repetition and variation (Marx, 1837; Reicha, 1814; Riemann, 1912;

Gissel Velarde · David Meredith
Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark
e-mail: {gv, dave}@create.aau.dk

Tillman Weyde
Department of Computer Science, City University London, London, UK
e-mail: t.e.weyde@city.ac.uk

D. Meredith (ed.), *Computational Music Analysis*,
DOI 10.1007/978-3-319-25931-4_12

Schoenberg, 1967), leading to the development of *paradigmatic analysis* by Ruwet (1966) and Nattiez (1986) during the latter half of the 20th century. Ruwet's method consists of an exhaustive similarity comparison of small units or segments in order to generate a structural description (see Monelle, 1992). Paradigmatic analysis focuses on clustering similar segments in a melody into "paradigms", regardless of where these segments might occur. It is typically carried out in parallel with *syntagmatic analysis* which focuses on identifying sequential relationships between consecutive segments. Syntagmatic and paradigmatic analysis can be seen as complementary tools for exploring the *semiotic* structure of a melody.

Almost three decades after the work by Ruwet, the first computational models to automate paradigmatic analysis of music appeared (Adiloglu et al., 2006; Anagnostopoulou and Westermann, 1997; Cambouropoulos, 1998; Cambouropoulos and Widmer, 2000; Conklin, 2006; Conklin and Anagnostopoulou, 2006; Grilo et al., 2001; Höthker et al., 2001; Weyde, 2001). However, it is difficult to evaluate these models, as some are not fully automated (e.g., require a user-supplied segmentation), the implementations are generally not public and they have not been tested on a common ground truth. Although the notion of defining a ground truth at all for a musical analysis is controversial, the MIREX task on discovery of repeated themes and sections (Collins, 2014) offers a practical opportunity to evaluate thematic analysis algorithms. However, it should be noted that the 'ground truth' analyses used in this task do not include any analyses by experts in paradigmatic analysis.

In this chapter, we focus on describing a fully automated method of musical analysis that closely resembles paradigmatic analysis. It has been implemented in Matlab and it is publicly available.[1] The method is based on segmenting melodies, clustering the resulting segments by similarity and then ranking the clusters obtained. In Sect. 12.3 we present the results obtained when our method was used for discovering repeating themes and sections in the Johannes Kepler University Patterns Development Database (JKU PDD).[2] We also compare these results with those obtained using other methods. In order to test the generalizability of the proposed method, we also evaluated it on a second musicological task, namely, that of identifying the parent compositions of excerpts from J. S. Bach's Two-Part Inventions (BWV 772–786).[3]

[1] Available at http://www.create.aau.dk/music/software/. It is implemented in MATLAB (R2014a, The Mathworks, Inc), using the following toolboxes: Signal Processing, Statistics, Symbolic Math, Wavelet, and the MIDI Toolbox (Eerola and Toiviainen, 2004). We also used an implementation of the dynamic time warping algorithm (DTW) by Paul Micó, accessed on 30-April-2013 from http://www.mathworks.com/matlabcentral/fileexchange/16350-continuous-dynamic-time-warping.

[2] https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip. Accessed on 12-May-2014.

[3] MIDI encodings edited by Steve Rasmussen,
http://www.musedata.org/encodings/bach/rasmuss/inventio/. Accessed April 2011

12.1.1 Melodic Structure and Wavelet Analysis

Our understanding of melodic structure has benefited from work that has been carried out in a number of fields, including music theory, psychology neuroscience and computer science. For example, melodic contour has been studied by Huron (1996), who classified melodies into 9 types according to their shapes (e.g., ascending, descending, arc-like, etc.) by considering the first, last and average pitches of a melody. In contrast, Schenkerian analysis aims to recursively reduce the musical surface or *foreground* to a *fundamental structure (Urzatz)* via one or more *middleground* levels *(Schichten)* (Schenker, 1935). Furthermore, listeners typically hear melodies to be "chunked" into *segments* or, more generally, *groups* (Cambouropoulos, 1997; Lerdahl and Jackendoff, 1983; Tenney and Polansky, 1980). Neuroscientific evidence from fMRI studies suggests that brain activity increases when subjects perceive boundaries between musical movements, and, indeed, boundaries between events in other, non-musical domains (Kurby and Zacks, 2008). Such evidence strongly supports the notion that segmentation is an essential component of perception, occurring simultaneously at multiple timescales. Psychological approaches focus on perception and memory and have tried to determine relevant melodic structures empirically (see, e.g., Lamont and Dibben, 2001; Müllensiefen and Wiggins, 2011b).

Computational approaches to the analysis of melodic structure include geometric approaches to pattern discovery, grammars, statistical descriptors, Gestalt features and data mining (see, e.g., Conklin, 2006; Mazzola et al., 2002; Meredith et al., 2002; Weyde, 2002). Wavelet analysis is a relatively new approach that has been widely used in audio signal processing. However, to our knowledge, it has been scarcely used on symbolic music representations, except by Smith and Honing (2008), who used wavelets to elicit rhythmic content from sparse sequences of impulses of a piece, and Pinto (2009), who used wavelets for melodic indexing as a compression technique.

As mentioned above, the wavelet-based method that we present below is closely related to paradigmatic analysis. It is based on the assumption that, if a melody is segmented appropriately, then it should be possible to produce a high-quality analysis by gathering together similar segments into clusters and then ranking these clusters by their importance or salience. In our study, we were particularly interested in exploring the effectiveness of the *wavelet transform* (WT) (Antoine, 1999; Farge, 1992; Mallat, 2009; Torrence and Compo, 1998) for representing relevant properties of melodies in segmentation, classification and pattern detection.

Wavelet analysis is a mathematical tool that compares a time-series with a wavelet at different positions and time scales, returning similarity coefficients. There are two main forms of the WT, the *continuous wavelet transform* (CWT) and the *discrete wavelet transform* (DWT). The CWT is mostly used for pattern analysis or feature detection in signal analysis (e.g., Smith and Honing, 2008), while the DWT is used for compression and reconstruction (e.g., Antoine, 1999; Mallat, 2009; Pinto, 2009). In our method, we sample symbolic representations of melodies or monophonic voices to produce one-dimensional (1D) *pitch signals*. We then apply the continuous wavelet transform (CWT) to these pitch signals, filtering with the Haar wavelet (Haar, 1910). Filtering with wavelets at different scales resembles the mechanism by

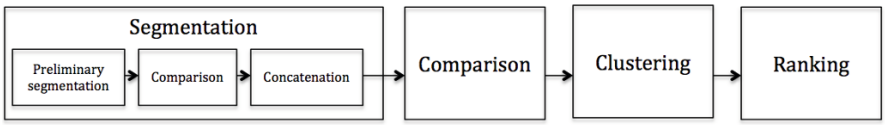

Fig. 12.1 A schematic overview of the main stages of the proposed method

which neurons, such as orientation-selective simple cells in the primary visual cortex, gather information from their receptive fields (Hubel and Wiesel, 1962). Indeed, more recently, Gabor wavelet pyramids have been used to model the perception of visual features in natural scenes (Kay et al., 2008).

Wavelet coefficient encodings seem to be particularly appropriate for melodic analysis as they provide a transposition-invariant representation. We also use wavelet coefficient representations to determine local segment boundaries at different time scales, which accords well with the notion that listeners automatically organize the musical surface into coherent segments, or groups, at various time scales (Lerdahl and Jackendoff, 1983).

12.2 Method

The method presented in this chapter extends our previously reported approach to melodic segmentation and classification based on filtering with the Haar wavelet (Velarde et al., 2013), and also incorporates an approach to segment construction similar to that developed by Aucouturier and Sandler (2002) for discovering patterns in audio data. A schematic overview of the method is shown in Fig. 12.1. In the following sub-sections we explain the method in detail.

12.2.1 Representation

A wide variety of different strategies have been adopted in music informatics for representing melodies, including (among others) viewpoints (Conklin, 2006), strings (McGettrick, 1997), contours (Huron, 1996), polynomial functions (Müllensiefen and Wiggins, 2011a), point sets (Meredith et al., 2002), spline curves (Urbano, 2013), Fourier coefficients (Schmuckler, 1999) and global features (van Kranenburg et al., 2013).

The representations used in this study are illustrated in Fig. 12.2. The top graph in this figure shows what we call a *raw pitch signal*. This is a discrete pitch signal, v, with length, L, constructed by sampling from MIDI files at a rate, r, in samples per quarter note (qn). MIDI files encode pitches as MIDI Note Numbers (MIDI NN). We denote the pitch value at time point t by $v[t]$. This representation is not used for

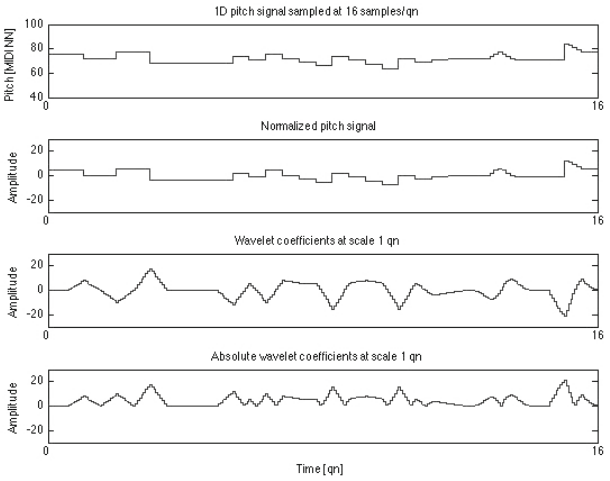

Fig. 12.2 Representations used in the method. From top to bottom: a raw pitch signal, a normalized pitch signal, a wavelet coefficient representation and an absolute wavelet coefficient representation

segment comparison directly. It is either filtered by the Haar wavelet or transformed into what we call a *normalized pitch signal* in order to obtain a transposition-invariant representation which is then segmented.

The second graph in Fig. 12.2 shows a *normalized pitch signal*, obtained by subtracting the average pitch of a segment from the pitch values in that segment. This process is applied to each segment individually after segmentation. It serves to reduce the measured dissimilarity between segments that have very similar contour but occur at different pitch heights (i.e., have different transpositions).

The third graph in Fig. 12.2 shows a *wavelet coefficient representation* resulting from carrying out a continuous wavelet transform (CWT) on the pitch signal with the Haar wavelet at a single time scale. This process tends to highlight structural features at the scale of the wavelet. The Haar wavelet (Haar, 1910) is used because it measures the movement direction of the melody and because its shape reflects the step-wise nature of symbolic pitch signals. Figure 12.3 shows an example of a Haar wavelet.

The CWT computed at a single time scale acts as a *filter* by the convolution of v, the pitch signal, with the scaled and flipped real-valued wavelet for each translation, u, and scale, s:

$$w_s[u] = \sum_{\ell=1}^{L} \psi_{s,u}[\ell] v[\ell] \; . \tag{12.1}$$

To avoid edge effects due to finite-length sequences (Torrence and Compo, 1998), we pad on both ends with a mirror image of v (Woody and Brown, 2007). To maintain

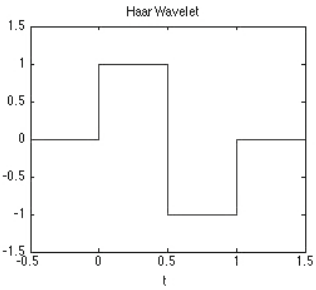

Fig. 12.3 The Haar wavelet

the signal's original length, the segments that correspond to the padding on both ends are removed after convolution.

The bottom graph in Fig. 12.2 shows an *absolute wavelet coefficient* representation. The value at each time point in this representation is the absolute value of the wavelet coefficient at that time point.

The type of wavelet to use depends on the kind of information one wishes to extract from the signal, since the wavelet coefficients combine information about the signal and the analysing function (Farge, 1992). We use the Haar wavelet (Haar, 1910) as the analysing function, as defined by Mallat (2009):

$$\psi_t = \begin{cases} 1, & \text{if } 0 \leq t < 0.5, \\ -1, & \text{if } 0.5 \leq t < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{12.2}$$

The choice of time scale depends on the scale of structure in which one is interested. Local structure is best analysed using short time scales, while longer-term structure can be revealed by using wavelets at longer time scales. When features of the wavelet-based representations are used for segmentation (as will be described in Sect. 12.2.2), using a shorter wavelet leads to smaller segments in general. We therefore expect shorter wavelets to be more appropriate for finding smaller melodic structural units such as motives, while longer wavelets might be expected to produce segments at longer time scales such as the phrase level and above. In the experiments reported below, we used a variety of different scales in order to explore the effect of time scale on performance.

12.2.2 Segmentation

Segmentation is a central component of music perception, occurring simultaneously at multiple timescales as an adaptive mechanism of the brain. It has been shown that brain activity increases transiently at musical movement boundaries, as well as other non-musical event boundaries (Kurby and Zacks, 2008). In agreement with the neuroscientific evidence, most theories of music perception and cognition note the

importance of segmentation, or grouping at various different time scales. Typically, such theories concentrate on the perceived associations of events, relating visual Gestalt principles to the musical domain. Examples of such theories include Tenney and Polansky's theory of temporal Gestalt-units (Tenney and Polansky, 1980), Lerdahl and Jackendoff's theory of grouping structure (Lerdahl and Jackendoff, 1983) and Cambouropoulos' Local Boundary Detection Model (LBDM) (Cambouropoulos, 1997, 2001). The rules in these models address changes in both local parameters and longer-term averages. Similarly, wavelet filters could be used to represent melodic movements at different scales, leading to different levels of localization on the time-axis for deriving group boundaries. Conklin (2006) also stresses the importance of melodic analysis on segmentation. He additionally demonstrates the effect of different symbolic melodic representations called *viewpoints* at different time scales (note, beat, bar, phrase and piece level) in the context of style discrimination.

As shown in Fig. 12.1, the *Segmentation* phase of our method is split into three subphases: *Preliminary segmentation*, *Comparison* and *Concatenation*. Each of these subphases will now be described.

12.2.2.1 Preliminary Segmentation

In this study, we explored three strategies for producing a preliminary segmentation: constant-duration segmentation; segmentation at zero crossings in the wavelet coefficient and absolute wavelet coefficient representations; and segmentation at local maxima in the absolute wavelet coefficient representation. The lower three graphs in Fig. 12.4 show three of the possible combinations of representation and segmentation.

The simplest segmentation strategy that we explore is *constant-duration segmentation* in which the signal is chunked into segments of constant duration (with the possible exception of the final segment which could be shorter than the other segments). The second graph in Fig. 12.4 shows an example of this type of segmentation combined with a normalized pitch signal representation.

We also experiment with *zero-crossings* segmentation in the wavelet-based representations, where segment boundaries are set at time points with value zero in the representation. Zero-crossings occur when the inner product between the melody and the Haar wavelet is zero. This means that the average pitch in the first half of the scale period is equal to the average pitch in the second half of the scale period.

The third segmentation strategy we use is *absolute maxima* segmentation, where segment boundaries are set at time points corresponding to local maxima in the absolute wavelet coefficient representation. These maxima occur when the inner product of the wavelet and the signal is locally maximal. In our case, this corresponds to time points when there is a maximal positive or negative correlation between the shape of the melody and the Haar wavelet. These points occur when there is a locally maximal fall or rise in average pitch content at the scale of the wavelet used. The absolute maxima of a real wavelet such as the Haar wavelet are a special case of the *modulus maxima* of a wavelet transform in general. The latter were used by Muzy

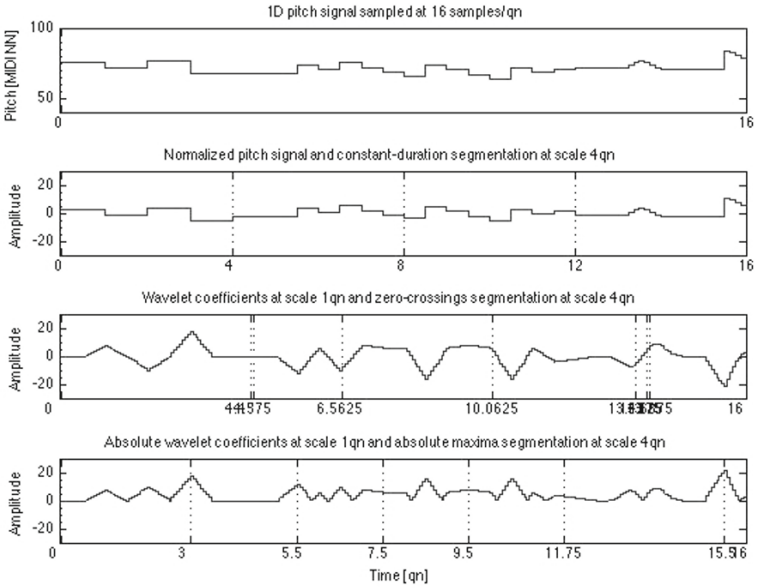

Fig. 12.4 Segmentation approaches used in the method, from top to bottom: a raw pitch signal without segmentation; normalized pitch signal and constant-duration segmentation at a scale of 4 qn; wavelet coefficient representation filtered with the Haar wavelet at a scale of 1 qn and segmented at zero-crossings at a scale of 4 qn; absolute wavelet coefficient representation filtered at a scale of 1 qn and segmented at absolute maxima at a scale of 4 qn. Note that the wavelet scales used to generate the *representations* shown in the third and fourth graphs are different from those used to produce the *segmentations*. The segmentation points therefore do not necessarily coincide with zero-crossings or maxima in the wavelet coefficient representations shown

et al. (1991) to show the structure of fractal signals and by Mallat and Hwang (1992) to indicate the location of edges in images. The bottom graph in Fig. 12.4 shows an example of absolute maxima segmentation of an absolute wavelet coefficient representation.

The segments obtained using these three strategies generally have different durations. However, in order to measure similarity between them using standard metrics such as *city block* or *Euclidean distance*, it is necessary for the segments to be the same length. We achieve this by defining a maximal length for all segments and padding shorter segments as necessary with zeros at the end.

12.2.2.2 Comparison

Segments are compared by building an $m \times m$ distance matrix, H, giving all pair-wise distances between segments in terms of normalized distance. m is the number of

segments. We use three different distance measures: *Euclidean distance*, *city block distance* and *dynamic time warping* (DTW). For city block and Euclidean distances, the segments compared must be of equal length and in these cases the normalization consists of dividing the pairwise distance by the length of the smallest segment before segment-length equalization by zero padding. When using DTW, which is an alignment-based method, it is not necessary to equalize the lengths of the segments being compared. In this case, therefore, the normalization consists of dividing the distance by the length of the aligned segments.

We use the *Euclidean distance* $d_{\mathrm{E}}(x,y)$ between two segments, x and y, which is defined as follows:

$$d_{\mathrm{E}}(x,y) = \sqrt{\sum_{j=1}^{n} (x[j]-y[j])^2}\ , \tag{12.3}$$

and the *city block distance* $d_{\mathrm{C}}(x,y)$ between x and y:

$$d_{\mathrm{C}}(x,y) = \sum_{j=1}^{n} |x[j]-y[j]|\ . \tag{12.4}$$

The *dynamic time warping distance* (DTW), $d_{\mathrm{D}}(x,y)$, is the minimal cost of a *warping path* between sequences x and y. A warping path of length, L, is a sequence of pairs $p = ((n_1,m_1),...,(n_L,m_L))$, where n_i is an index into x and m_i is an index into y. p needs to satisfy several conditions which ensure that it can be interpreted as an alignment between x and y that allows skipping elements in either sequence (see Müller, 2007, p. 70). The DTW distance, $d_{\mathrm{D}}(x,y)$, is then defined to be the total cost of a warping path, defined to be the sum of a local cost measure, $c(x[n_i],y[m_i])$, along the path:

$$d_{\mathrm{D}}(x,y) = \sum_{i=1}^{L} c(x[n_i],y[m_i])\ , \tag{12.5}$$

where, here, $c(x[n_i],y[m_i])$ is defined to be simply the absolute difference, $|x[n_i] - y[m_i]|$.

Having computed all the pairwise distances in the matrix, H, these values are then normalized in the range $[0,1]$ by dividing each pairwise distance by the largest distance in the matrix for that distance type.

12.2.2.3 Concatenation of Segments

The final subphase of the segmentation phase is to concatenate consecutive segments found in the preliminary segmentation to form larger units that are then compared, clustered and ranked in the subsequent phases of the method.

The first subphase of the segmentation phase gives a preliminary segmentation of the melody. It is preliminary, as it may be the case that a repeated (or approximately repeated) segment discovered in the preliminary segmentation only occurs as part of a longer repeated segment, such that a paradigmatic relation is found. In such

cases, one would generally only be interested in the longer repeated segment (this relates to the concept of "closed patterns" (see Lartillot, 2005, and Chap. 11, this volume) and Meredith et al.'s (2002) concept of "maximal translatable patterns" (see also Chap. 13, this volume). One would only want to report the shorter segment if it also occurred independently of the longer segment. In the third subphase of the segmentation phase, we therefore concatenate, or merge locally, the preliminary segments derived in the preliminary segmentation into generally longer units, that are then passed on to the later phases of the method.

Segments are concatenated based on their similarity. We therefore set a threshold, τ, that defines the level of similarity between preliminary segments required to allow concatenation. The $m \times m$ distance matrix, H, is therefore binarized as follows:

$$H(i,j) = \begin{cases} 1, & \text{if } H(i,j) \leq \tau, \\ 0, & \text{otherwise,} \end{cases} \tag{12.6}$$

for $1 \leq i \leq m$ and $i \leq j \leq m$ (note that we use 1-based indexing in this chapter).

Segments are concatenated to form units based on the information contained in the upper triangle including the leading diagonal in the binarized similarity matrix, H, scanning the matrix horizontally and diagonally. A *unit*, $\overline{(i,j)}$, $i \leq j$, consists of the concatenated segments $i, \ldots, j$, and we use two concatenation processes to generate units.

A process of *horizontal concatenation* generates units that consist of consecutive occurrences of the "same" segment (i.e., corresponding to horizontal sequences of consecutive 1s in the binarized similarity matrix, H). The units, $\overline{(i,k)}$, generated by this process are those for which $hor(i,k)$ is true, where

$$hor(i,k) \iff (hor(i,k-1) \wedge H(k-1,k) = 1) \vee (i = k)\,. \tag{12.7}$$

A process of *diagonal concatenation* generates units that are repeated in the piece, and $dia(i,j)$ must be true, where

$$\begin{aligned} dia(i,j) \iff & (dia(i,j-1) \wedge \exists \ell, k \mid \ell - k = j - i \wedge dia(k,\ell-1) \wedge H(j-1,\ell-1) = H(j,\ell) = 1) \\ & \vee (j - i = 1 \wedge \exists \ell \mid H(i,\ell-1) = H(j,\ell) = 1)\,. \end{aligned} \tag{12.8}$$

Any $hor(i,j)$ or $dia(i,j)$ that is not a strict subset of another generates a unit $\overline{(i,j)}$. Subsets will be identified as *trivial units*.

When these two concatenation processes are carried out on the matrix in Fig. 12.5, horizontal concatenation generates the unit $\overline{(9,10)}$ and diagonal concatenation generates the units $\overline{(1,2)}$, $\overline{(4,5)}$ and $\overline{(7,8)}$.

The concatenation method presented here is similar to the one described by Aucouturier and Sandler (2002).

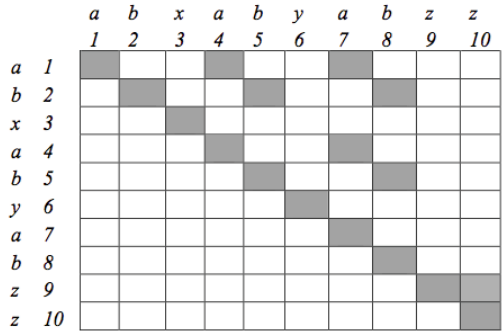

Fig. 12.5 Upper triangular matrix, grey means 1 and white 0. It corresponds to the binarized distance matrix H of the sequence $v_1 = abxabyabzz$

12.2.3 Comparison and Clustering of Units

In this second comparison, the units constructed in the previous concatenation step (Sect. 12.2.2.3) are compared using the same process of similarity measurement as that described in Sect. 12.2.2.2. Any two units $\overline{(\ell, j)}$ and $\overline{(p, r)}$ obtained by concatenation, will then be units x and y respectively, to be compared in this second comparison.

Having obtained values for the pairwise similarity between units, these similarity values are then used to cluster the units into classes. To achieve this, we use a simple hierarchical agglomerative clustering method called *single linkage*, or *nearest-neighbour*, which produces a series of successive fusions of the data, starting from N single-member clusters that fuse together to form larger clusters (Everitt et al., 2011; Florek et al., 1951; Johnson, 1967; Sneath, 1957). Here, the distance matrix obtained from the comparison as described in Sect. 12.2.3 is used for clustering. *Single linkage* takes the smallest distance between any two units, one from each group or cluster. The distance $D(X,Y)$ between clusters X and Y is described as

$$D(X,Y) = \min_{x \in X, y \in Y} d(x,y)\,, \tag{12.9}$$

where clusters X and Y are formed by the fusion of two clusters, x and y, and $d(x,y)$ denotes the distance between the two units x and y (Everitt et al., 2011). Consider the case of five units or clusters v, w, x, y and z, as shown on the left in Fig. 12.6 as points in a Euclidean space. The minimal distance occurs for x and y, and for z and w. Then, two new clusters are formed, a cluster s consisting of x and y and a cluster t consisting of z and w. The next minimal distance occurs for v and t, forming a new cluster u consisting of v and t. Finally, clusters s and u are grouped together into a cluster c. The right plot in Fig. 12.6 shows a dendrogram of the formed clusters. The y-axis corresponds to the distances between clusters; for instance, clusters x and y

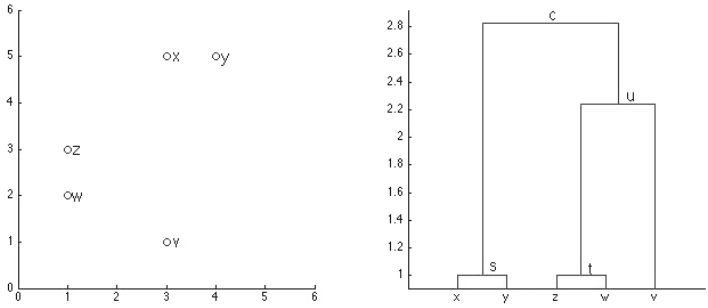

Fig. 12.6 Example of the hierarchical clustering of units or clusters *v*, *w*, *x*, *y* and *z*. Left plot shows the units in a Euclidean space. Right plot shows a dendrogram of the formed clusters

have a distance of 1, and clusters *t* and *u* have a distance of 2.2. In this example, the number of clusters ranges from 1, where all units form a single cluster, to 5, where each cluster contains just one unit. The number of clusters can be set to be three, having clusters *s*, *t* and *u* or it can be set to two, giving clusters *s* and *u*. Finally, the number of clusters is set to yield the best classification results.[4]

12.2.4 Ranking

In general, if X and Y are two parts of some object, then one can describe $X \cup Y$ in an *in extenso* fashion simply by specifying the properties of each atomic component in X and Y. Alternatively, if there exists a sufficiently simple transformation, T, that maps X onto Y, then it may be possible to provide a compact description of $X \cup Y$ by providing an in extenso description of X along with a description of T.[5]

In the current context, each cluster generated by the previous stage of the method contains units (i.e., parts of a melody) that are similar to each other. If every member of a cluster can be generated by a simple transformation of one member (e.g., if all the units within a cluster are exact repeats of the first occurrence), then the portion of the melody *covered* by the cluster (i.e., the union of the units in the cluster) can be represented by giving an explicit description of the first occurrence along with the positions of the other occurrences. If the members of the cluster do not overlap, then such a representation can be compact because the starting position of a unit can usually be specified using fewer bits than explicitly describing the content of the unit. This would give a losslessly compressed encoding of the part of the melody

[4] When preliminary experiments were performed on the JKU PDD, using between 3 and 10 clusters, the best classification results were obtained using 7 clusters. We therefore used 7 clusters in the experiments reported in Sect. 12.3 below.

[5] This idea is discussed in more detail in Chap. 13, this volume.

covered by the union of the units in the cluster. This is the essential idea behind the compression-driven geometric pattern discovery algorithms described by Meredith (2006, 2013) and Forth (2012). If we represent the music to be analysed as a set of points in pitch-time space and if a cluster (or 'paradigm'), C, only contains the *exact* occurrences of a pattern, p, then the compression ratio achieved is

$$CR(C) = \frac{|\bigcup_{q \in C}\{q\}|}{|p| + |C| - 1}\,, \tag{12.10}$$

where $|\cdot|$ denotes the cardinality of a set. Here, however, the units within a cluster are not necessarily exact repetitions of some single pattern. This means that the degree of compression achievable with one of the clusters generated in the previous sections will not, in general, be as high as in (12.10).

Collins et al. (2011) have provided empirical evidence that the compression ratio achievable in this way by a set of occurrences of a pattern can be used to help predict which patterns in a piece of music are heard to be noticeable and/or important. In the method presented in this chapter, we therefore adapt (12.10) to serve as a measure of importance or noticeability for the clusters generated in the previous phase of the method. Here, we define the "compression ratio", CR_k, of cluster k as follows:

$$CR_k = \frac{\sum_{i=1}^{n_k} S_i}{(n_k + \bar{S}_k)}\,, \tag{12.11}$$

where n_k is the number of units in cluster k, S_i is the length in sample points of unit i in cluster k, and $\bar{S}_k$ is the mean length of a unit in cluster k. Clusters are ranked into descending order by this value of "compression ratio". All clusters are kept in the final output.

12.3 Experiments

The method described above was evaluated on two tasks: discovering repeated themes and sections in monophonic music; and identifying the parent works of excerpts from J. S. Bach's Two-Part Inventions (BWV 772–786). The methods used and results obtained in these experiments will now be presented.

12.3.1 Experiment 1: Discovering Repeated Themes and Sections in Monophonic Music

Various computational methods for discovering patterns in music have been developed over the past two decades (see Janssen et al., 2013, for a recent review), but only recently have attempts been made to compare their outputs in a rigorous way.

Notable among such attempts are the two tasks on discovering repeated themes and sections that have been held at the Music Information Retrieval Evaluation eXchange (MIREX) in 2013 and 2014 (Collins, 2014). In these tasks, algorithms have been run on a set of five pieces and the analyses generated by the algorithms have been compared with ground truth analyses by expert analysts. A number of measures were devised for evaluating the performance of pattern discovery algorithms in this competition and comparing the output of an algorithm with a ground truth analysis (Collins, 2014). Collins has also provided a training database, the JKU PDD, which exists in both monophonic and polyphonic versions. The JKU PDD consists of the following five pieces along with ground truth analyses:

- Orlando Gibbons' madrigal, "Silver Swan" (1612);
- the fugue from J. S. Bach's Prelude and Fugue in A minor (BWV 889) from Book 2 of *Das wohltemperirte Clavier* (1742);
- the second movement of Mozart's Piano Sonata in E flat major (K. 282) (1774);
- the third movement of Beethoven's Piano Sonata in F minor, Op. 2, No. 1 (1795); and
- Chopin's Mazurka in B flat minor, Op. 24, No. 4 (1836).

The monophonic versions of the pieces by Beethoven, Mozart and Chopin were produced by selecting the notes in the most salient part (usually the top part) at each point in the music. For the contrapuntal pieces by Bach and Gibbons, the monophonic encodings were produced by concatenating the voices (Collins, 2014).

We used the JKU PDD as a training set for determining optimal values for the parameters of the analysis method described above. Heuristics based on knowledge gained from previous experiments (Velarde et al., 2013) were used to start tuning the parameters. Then, in an attempt to approach optimal values, all parameters were kept fixed, except one which was varied along a defined range to find an optimal adjustment. This process was repeated for all parameters. Finally, the method was run on the JKU PDD with 162 different parameter value combinations, consisting of all possible combinations of the following:

- 1 sampling rate: 16 samples per qn
- 3 representations: normalized pitch signal, wavelet coefficients filtered at the scale of 1 qn, absolute wavelet coefficients filtered at the scale of 1 qn
- 3 segmentation strategies: constant-duration segmentation, segmentation at zero-crossings, segmentation at absolute maxima
- 2 scales for segmentation: 1 qn and 4 qn
- 1 threshold for binarizing the similarity matrix: 0.001
- 3 distances for measuring similarity between segments on the first comparison: city block (CB), Euclidean (Eu) and dynamic time warping (DTW)
- 3 distances for measuring similarity between segments on the second comparison: city block (CB), Euclidean (Eu) and dynamic time warping (DTW)
- 1 strategy for equalizing the lengths of segments for comparison: segment length normalization by zero padding
- 1 clustering method: Single linkage (nearest neighbour)
- 1 value for the number of clusters: 7

- 1 criterion for ranking clusters: compression ratio

12.3.1.1 Results

We used the monophonic version of the JKU PDD with the evaluation metrics defined by Collins (2014) and Meredith (2015), which we computed using Collins' Matlab implementation.[6] The evaluation metrics consist of a number of variants on standard precision, recall and F_1 score, designed to allow algorithms to gain credit for generating sets of occurrences of patterns that are similar but not identical to those in the ground truth. The standard versions of the metrics are not adequate for evaluating pattern discovery algorithms because they return 0 for a computed pattern even if it differs from a ground truth pattern by only one note.

The more robust versions of the precision, recall and F_1 score are designed to measure (1) the extent to which an algorithm finds at least one occurrence of a pattern (*establishment recall/precision/F_1 score*); (2) the extent to which an algorithm finds all the occurrences of a pattern (*occurrence recall/precision/F_1 score*); and (3) the overall similarity between the set of occurrence sets generated by an algorithm and the set of occurrence sets in a ground truth analysis (*three-layer precision/recall/F_1 score*). As these different metrics reveal different aspects of the method's strengths or weaknesses, we decided to evaluate our method based on the standard F_1 score, where P is precision and R is recall

$$F_1 = \frac{2PR}{P+R} \tag{12.12}$$

and on the mean of establishment F_1 $(F1\_est)$, occurrence F_1 at (c=.75) $(F1\_occ_{(c=.75)})$, occurrence F_1 at (c=.5) $(F1\_occ_{(c=.5)})$ (Collins, 2014), and three-layer F_1 $(F1\_TL)$ (Meredith, 2015):

$$F1\_mean = \frac{F1\_est + F1\_occ_{(c=.75)} + F1\_occ_{(c=.5)}) + F1\_TL}{4}. \tag{12.13}$$

Figure 12.7 shows the highest mean F_1 scores ($F1\_mean$) for each combination, considering segmentation scale, representation type and segmentation type. The left plot shows nine combinations where the segmentation scale was 1 qn, while the right plot shows the scores of nine combinations where the segmentation scale was 4 qn. For each plot in Fig. 12.7, there are 3 bars grouped for each segmentation method, where the grey tones (dark grey, light grey and white) indicate the three representation types, and finally, the distance measures associated with the first and second comparison (e.g., "EU,EU", "CB,CB", etc.). Figure 12.8 shows the corresponding standard F_1 scores for the same combinations. Finally, Fig. 12.9 shows the runtimes in seconds obtained with our implementations of the method, associated with each combination.

[6] https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip. Accessed on 12-May-2014.

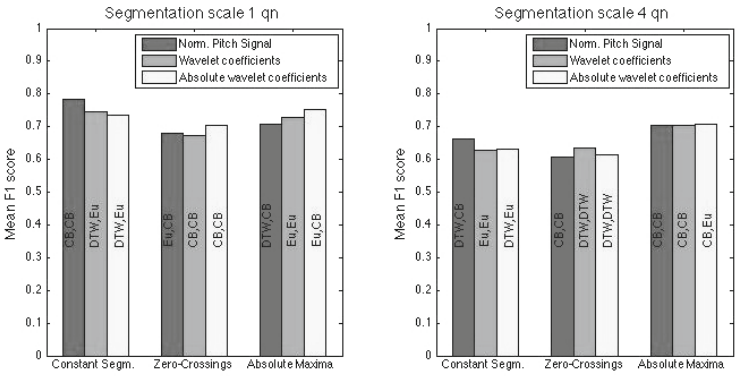

Fig. 12.7 Mean F_1 score (*F1_mean*)

We ran the experiment twice, the first time keeping trivial units and the second time discarding trivial units. Figures 12.7, 12.8 and 12.9 show the results when keeping trivial units. A Wilcoxon signed rank test indicated that keeping or discarding trivial units did not significantly affect the results of mean F_1 scores ($Z = -1.2439$, $p = 0.2135$), standard F_1 scores ($Z = -1.633$, $p = 0.1025$), or runtimes ($Z = -0.8885$, $p = 0.3743$), for a segmentation scale of 1 qn. Similarly, no difference was found in the results when keeping or discarding trivial units for a scale of 4 qn for mean F_1 scores ($Z = 1.007$, $p = 0.3139$), standard F_1 scores ($Z = 0$, $p = 1$), or runtimes ($Z = -0.53331$, $p = 0.5940$). Therefore, only the results of the first run are shown and explained in the following paragraphs.

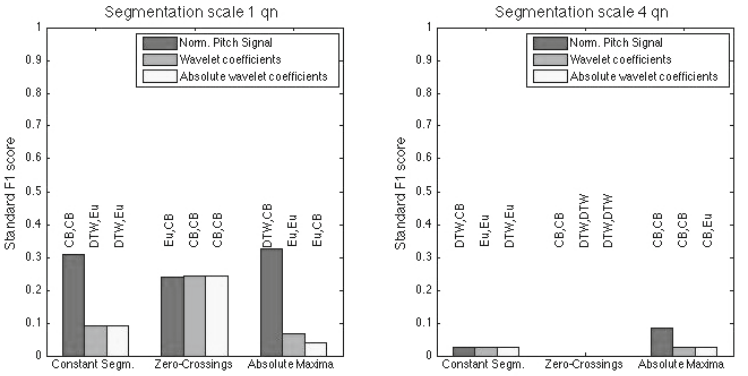

Fig. 12.8 Standard F_1 score

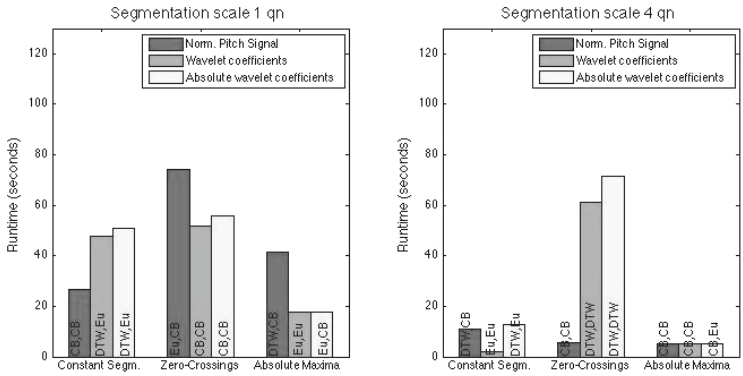

Fig. 12.9 Runtimes in seconds obtained using our implementation of the method. The implementation was programmed using Matlab 2014a and run on a MacBook Pro using MAC OS X with a 2.3 GHz, Intel Core i7 processor and 8 GB 1600 MHz DDR3 RAM

According to the parameters tested, we observe that the segmentation scale used in the preliminary segmentation phase has a greater effect on the results. Figures 12.8 and 12.9 show that using a smaller segmentation scale of 1 qn as opposed to 4 qn was in general slower but produced better results. A Wilcoxon signed rank test indicated there is a statistically significant difference between the use of a smaller and larger scale ($Z = 2.6656$, $p = 0.007$), suggesting that a scale of 1 qn should be used in the preliminary segmentation phase, for higher (mean or standard) F_1 scores.

In terms of mean F_1 score (Fig. 12.7), the normalized pitch signal representation worked slightly better than the wavelet representations when constant-duration segmentation was used. We speculate that only with additional pattern data containing greater variation between occurrences would the benefit of wavelet over normalized pitch representations emerge (see Sect. 12.3.2.1 for more discussion on this point). DTW was used less frequently than Euclidean or city block distance in the best-performing combinations. It seems possible that DTW might have proved more useful if the input representations had included temporal deviations such as *ritardando* or *accelerando* such as might occur in an encoding generated from a live performance.

From Figs. 12.7, 12.8 and 12.9 it is not possible to determine whether the running time is more dependent on the segmentation approach or on the distance measure used. Tables 12.1 and 12.2, show the highest mean F_1 scores of combinations using the same distance measure for both comparison phases, averaged by representation approach. From Table 12.1, it is possible to observe that when using a scale of 1 qn for the preliminary segmentation phase, Euclidean and city-block distances have similar performance, and their F_1 scores are higher than the ones delivered when using DTW distance. However, this gap becomes smaller when the scale is 4 qn. The results in Table 12.2 show that the running times using DTW are more than 8 times slower than those obtained using Euclidean or city-block distances. Evaluating runtimes according to segmentation approaches, it is possible to observe that for the smaller

Table 12.1 Mean F_1 scores averaged over representations, combinations of same distance measure for both comparisons. The rows correspond to the different combinations of distances (CB = city-block, Eu = Euclidean and DTW = dynamic time warping), while the columns correspond to the segmentation approaches (CS = constant-duration segmentation, ZC = zero-crossings segmentation, and AM = absolute maxima segmentation). Mean and standard deviation values are shown per row and per column

	Segmentation scale 1 qn					Segmentation scale 4 qn				
	CS	ZC	AM	**Mean**	SD	CS	ZC	AM	**Mean**	SD
CB-CB	0.74	0.69	0.75	**0.73**	0.03	0.65	0.60	0.70	**0.65**	0.05
Eu-Eu	0.73	0.68	0.72	**0.71**	0.03	0.63	0.59	0.70	**0.64**	0.05
DTW-DTW	0.57	0.64	0.60	**0.60**	0.04	0.59	0.61	0.66	**0.62**	0.03
Mean	**0.68**	**0.67**	**0.60**			**0.62**	**0.60**	**0.69**		
SD	0.10	0.03	0.08			0.03	0.01	0.03		

Table 12.2 Corresponding mean running times in seconds of the combinations in Table 12.1

	Segmentation scale 1 qn					Segmentation scale 4 qn				
	CS	ZC	AM	**Mean**	SD	CS	ZC	AM	**Mean**	SD
CB-CB	24.3	60.8	17.9	**34.32**	23.17	2.2	5.4	5.2	**4.23**	1.80
Eu-Eu	24.4	57.1	17.8	**33.10**	21.02	2.1	5.3	5.1	**4.16**	1.79
DTW-DTW	664.4	2248.2	720.1	**1210.91**	898.77	21.5	61.5	67.4	**50.14**	25.01
Mean	**237.69**	**788.70**	**251.93**			**8.58**	**24.04**	**25.92**		
SD	369.56	1263.98	405.44			11.17	32.44	35.97		

Table 12.3 Mean F_1 scores averaged over representations, when the concatenation phase is not performed. The rows of the Table indicate the distances used for comparison (CB = city-block, Eu = Euclidean and DTW = dynamic time warping), while the columns correspond to the segmentation approaches (CS = constant-duration segmentation, ZC = zero-crossings segmentation, and AM = absolute maxima segmentation). Mean and standard deviation values are shown per rows and per columns

	Segmentation scale 1 qn					Segmentation scale 4 qn				
	CS	ZC	AM	**Mean**	SD	CS	ZC	AM	**Mean**	SD
CB	0.10	0.18	0.11	**0.13**	0.04	0.22	0.23	0.18	**0.21**	0.03
Eu	0.10	0.14	0.10	**0.11**	0.02	0.22	0.21	0.16	**0.20**	0.03
DTW	0.10	0.09	0.11	**0.10**	0.01	0.22	0.20	0.18	**0.20**	0.02
Mean	**0.10**	**0.14**	**0.11**			**0.22**	**0.21**	**0.18**		
SD	0.00	0.04	0.01			0.00	0.02	0.01		

scale of 1 qn in the preliminary segmentation phase, the runtimes of constant-duration segmentation and wavelet absolute maxima segmentation are similar and about twice as fast as the runtimes of the zero-crossings segmentation. On the other hand, for a larger scale of 4 qn in the preliminary segmentation phase, constant-duration segmentation is three times faster than wavelet segmentation approaches.

Table 12.3 shows the effect of not using the concatenation phase: melodies undergo the preliminary segmentation phase, but skip the first comparison and the concatenation phases, such that all preliminary segments are used for the comparison, clustering and ranking phases. The results in Table 12.3 show that omitting the concatenation phase severely reduces the performance of the method on this task. In this

case, when segments are not concatenated, a segmentation scale of 4 qn is, in almost all combinations, twice as good as a segmentation scale of 1 qn. On the other hand, as seen in Table 12.1, a preliminary segmentation phase with a finer segmentation scale, helps to improve the identification of patterns in this dataset.

12.3.1.2 Comparison with Other Computational Methods

The other computational methods addressing the MIREX task on Discovery of repeated themes and sections, included geometric approaches (Meredith, 2013), incremental mining methods (Lartillot, 2014) and methods based on audio techniques (Nieto and Farbood, 2013, 2014).[7] For comparison, we selected our submission VM1, as this configuration was also selected for comparison in the published results of the task. The details of the parameters settings of VM1 are described by Velarde and Meredith (2014).

Table 12.4 shows the results obtained by the different algorithms in the 2014 MIREX task on the monophonic version of the JKU Patterns Test Database (PTD). As can be seen in this table, our method ranked highest at discovering at least one occurrence of each ground truth pattern ($F1\_est$) as well as being the fastest method. Lartillot's method (OL1) performed better at finding inexact occurrences of patterns ($F1\_occ_{(c=.75)}$) but is considerably slower. VM1 and OL1 performed at a similar level with respect to finding exact occurrences of the patterns, and, in both cases, the standard deviation was high. The addition of more pieces to training and test databases over time will enable researchers to investigate the generalizability of their methods.

Table 12.4 Results on the JKU test set. NF1 (Nieto and Farbood, 2014), OL1 (Lartillot, 2014), VM1 (Velarde and Meredith, 2014) and DM10 (Meredith, 2013)

		$F1\_est$	$F1\_occ_{(c=.75)}$	TL_F1	$F1$	*Runtime*
NF1	Mean	0.50	0.41	0.33	0.02	480.80
	SD	0.14	0.27	0.12	0.05	558.43
OL1	Mean	0.50	**0.81**	0.43	0.12	35508.82
	SD	0.17	0.12	0.13	0.13	52556.11
VM1	Mean	**0.73**	0.60	**0.49**	**0.16**	**100.80**
	SD	0.14	0.09	0.14	0.15	119.18
DM10	Mean	0.55	0.62	0.43	0.03	161.40
	SD	0.06	0.09	0.08	0.04	194.87

[7] Results of the annual MIREX competitions on Discovery of Repeated Themes and Sections can be found on the MIREX website at at http://www.music-ir.org/.

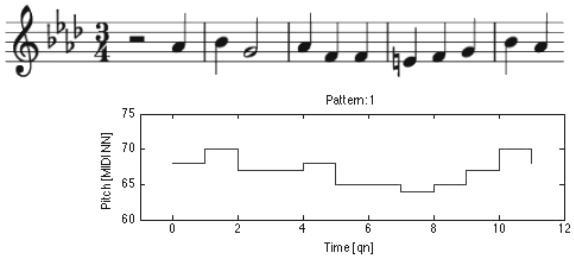

Fig. 12.10 Notation and pitch-signal representations of the first ground truth pattern for the third movement of Beethoven's Piano Sonata in F minor, Op. 2, No. 1 (1795)

12.3.1.3 Comparing Patterns Discovered Automatically with Patterns Identified by Experts

In this section, we present the output of the computational method compared to the JKU PDD ground truth analysis of the monophonic version of the third movement of Beethoven's Piano Sonata in F minor, Op. 2, No. 1 (1795). In order to visualize the ground truth and computationally discovered patterns and their occurrences, we will present them as pitch signals rather than in notation. To help with understanding the correspondence between the pitch signal representation and notation, Fig. 12.10 shows both representations of the first ground truth pattern.

The ground truth analysis for this piece identifies seven patterns and their occurrences as shown in Fig. 12.11. In this figure, plots on the left correspond to patterns, while plots on the right correspond to pattern occurrences. Each pattern occurrence is marked with vertical dotted lines in the graphs on the right side of the figure. All pitch signals have been shifted to start at time 0. The patterns are ordered, from top to bottom, in decreasing order of salience. The lengths of these seven ground truth patterns range from 12 to 119 qn. Some occurrences of the patterns overlap as is the case for the occurrences of pattern 1 and pattern 3, or pattern 2 and pattern 5.

The computational analysis of the piece can be seen in Fig. 12.12. The parameters used are the following:

- 1 sampling rate: 16 samples per qn
- representations: absolute wavelet coefficients filtered at the scale of 1 qn
- segmentation at absolute maxima
- scales for segmentation: 1 qn
- threshold for binarizing the similarity matrix: 0.001
- distance for measuring similarity between segments on the first comparison: city block (CB)
- distance for measuring similarity between segments on the second comparison: city block (CB)
- clustering method: Single linkage (nearest neighbour)
- value for the number of clusters: 7

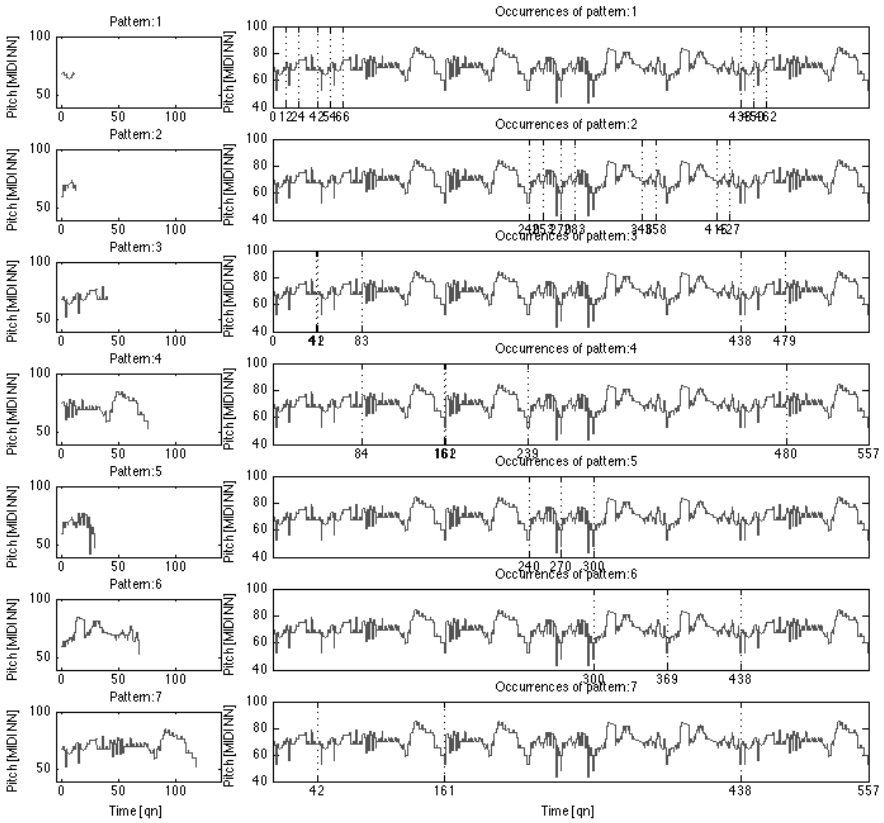

Fig. 12.11 JKU PDD Ground truth patterns for the third movement of Beethoven's Piano Sonata in F minor, Op. 2, No. 1 (1795). Pitch signal representation, with signals shifted to start at time 0. Plots on the left correspond to the patterns, while plots on the right correspond to the entire piece, with each pattern occurrence marked with a vertical dotted line at its starting and ending position

- criterion for ranking clusters: compression ratio

In this example, the number of clusters is the same as the number of patterns in the ground truth. Once again, the salience of patterns can be seen from top to bottom, where the most salient pattern is shown in the top plot. Six out of seven pattern shapes match approximately the ground truth pattern shapes (in some cases, some notes may be missing at the beginning or end of a pattern). The pattern that has been ranked as the most salient, corresponds to pattern 2 in the ground truth analysis, and all its four occurrences have been found. The shape of the second most salient computed pattern, does not resemble the shape of any of the patterns in the ground truth. Pattern 2 is a short-duration pattern, whose cluster contains several melodic units, including segments that approximate the occurrences of pattern 1 in the ground truth (this cannot be seen in Fig. 12.12). The remaining computed pattern

shapes (patterns 3–7) can be found in the ground truth, each with the same number of occurrences. The ranking of salience is not exactly the same as in the ground truth, but it is similar in chunks, such that:

- the first two computed clusters correspond to the first two pattern occurrences in the ground truth;
- computed cluster 3 corresponds to the occurrences of ground truth pattern 3;
- computed clusters 4–6 correspond to the occurrences of ground truth patterns 4–6,
- and finally the last computed cluster corresponds to the occurrences of the last ground truth pattern.

The second cluster contains several melodic units. In future work, we would like to cluster such clusters until they satisfy a given condition and discard clusters that fail to satisfy the condition. We expect that the effect on such clusters of keeping or discarding trivial units may be more evident if we carry out this process.

12.3.2 Experiment 2: Classification of Segments from J. S. Bach's Two-Part Inventions

We also evaluated the method on a second task where the goal was to recognize the parent works of excerpts from J. S. Bach's 15 Two-Part Inventions (BWV 772–786). In contrast to the first experiment, in this task, all segments were used in the evaluation, not just concatenated units. Also, whereas in the first experiment there was room for disagreement about the validity of the ground truth, in this second task, the ground truth was not controversial—there was no doubt as to which parent Invention each test excerpt belonged to. The notion that the piece to which an excerpt belongs can be identified on the basis of the content of the excerpt is based on the premise that the musical material in the excerpt is motivically related to the rest of the piece. Specifically, in the case of Bach's Two-Part Inventions, it is well established that the opening exposition of each of these pieces presents the motivic material that is developed throughout the rest of the piece, which is typically divided into three sections (Dreyfus, 1996; Stein, 1979). In this experiment, we followed the experimental setup described by Velarde et al. (2013), building the classifier from the expositions of the pieces and the test set from the three following sections of each piece. More precisely, an initial, 16 qn segment from each piece was used to build the classifier, and the remainder of each piece was split into three sections of equal length which were used to build the test set. We could have attempted to determine the length of each exposition precisely, but we wanted to avoid making subjective analytical judgements. We therefore used a fixed length of 16 qn as the length of each "exposition" section despite the fact that the actual lengths of the expositions in the Inventions vary. This particular length was chosen because it was the length of the longest exposition in the pieces, thus ensuring that no exposition material would be included in the test set.

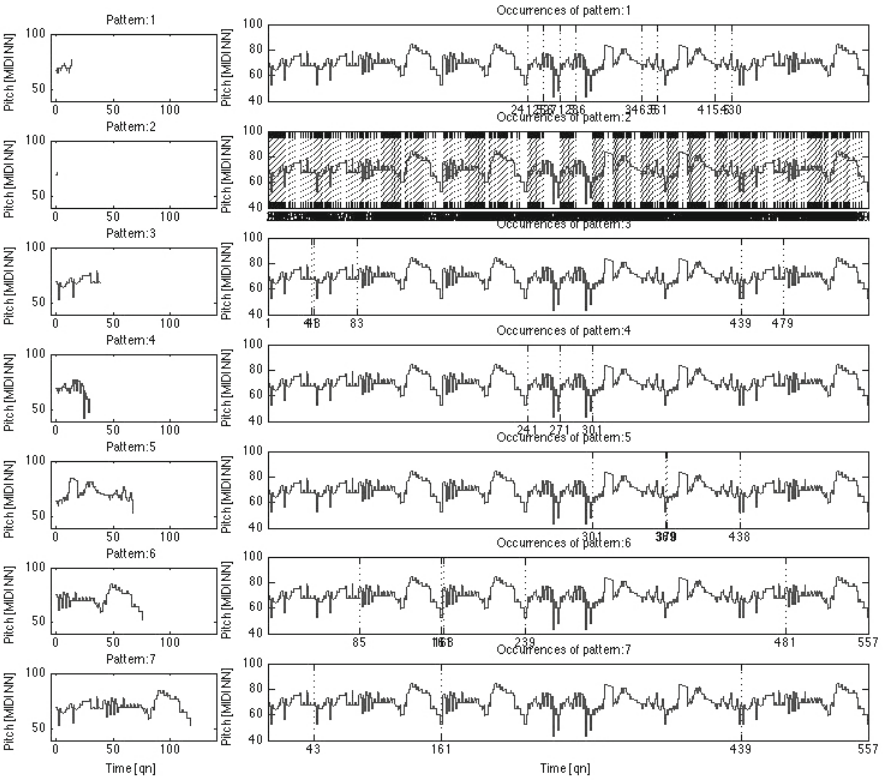

Fig. 12.12 Patterns discovered by the method for the third movement of Beethoven's Piano Sonata in F minor, Op. 2, No. 1 (1795), JKU PDD monophonic version. Pitch signal representation, with signals shifted to start at time 0. Plots on the left correspond to the patterns, while plots on the right correspond to the entire piece, with each pattern occurrence marked with a vertical dotted line at its starting and ending position

We were also interested in investigating the amount of initial expository material required to enable the parent works of excerpts to be accurately identified. We therefore constructed classifiers from the first 4, 8 and 16 qn of the pieces.

Figure 12.13 shows schematically how the classifiers and the test sets were constructed. The classifier set C was built from segments $sc_{i,j}$ from the expositions of the 15 *Inventions*, where each segment could be from either the upper or the lower voice. $sc_{i,j}$ is the jth segment in Invention i. Each test set T was built from segments st, where each st could be from either the upper or the lower voice. We denote the jth segment in Invention i by $st_{i,j}$. To classify a segment st to one of the 15 classes, we applied 1-nearest neighbour classification (Mitchell, 1997). That is, we computed the distances between st and all sc in C, and classified st to the class i of the $sc_{i,j}$ that had the smallest distance to st. Each test excerpt was assigned the class most frequently

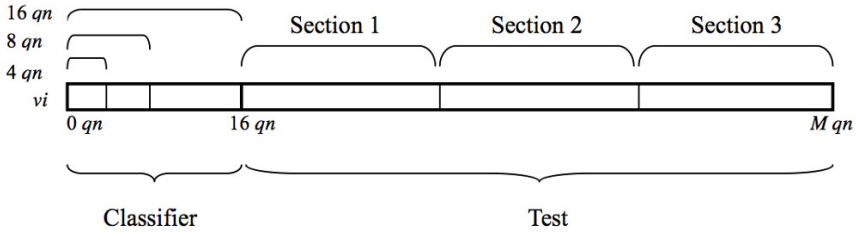

Fig. 12.13 Scheme of classifier and test construction based on signal v_i

predicted by its segments. In both cases we used the next nearest neighbour to break ties.

We expected higher classification rates with classifiers built from more exposition material, similar performance for the different combinations of wavelet-base classifiers, and higher classification rates in the first section compared to the following two, as the subject appears in the first section following the exposition at least once in each part (Stein, 1979).

The following parameters were used in the experiment:

- Sampling rate: 8 samples per qn[8]
- Representation: normalized pitch signal (WR), wavelet coefficients filtered at the scale of 1 qn (WR) and absolute wavelet coefficients filtered at the scale of 1 qn (WRA)
- Segmentation: constant-duration segmentation (CS), wavelet zero-crossing (ZC) and wavelet absolute maxima (AM)
- Scale segmentation at 1 qn
- Segment length normalization by zero padding
- Clustering: 1-nearest neighbour
- Distance measure: city block

12.3.2.1 Results

Figures 12.14 and 12.15 show the classification accuracy on each section, with the concatenation phase omitted and included, respectively. Both figures show the effect of segmentation and representation (columns vs. rows), and the number of qn used for the classifiers (asterisk, square, and circle markers). As expected, the amount of material used from the exposition (4, 8, or 16 qn) affects the classification success rates: the more material used, the higher the success rates. Moreover, segmentation has a stronger effect on the classification than representation. With respect to the results between sections, the classification rates for the first section are higher than

[8] The sampling rate was chosen to be the same as that used by Velarde et al. (2013).

those for the second and third sections. Representations associated with constant-duration segmentation are accurate in the first section after the exposition, where the subject is presented at least once in one of the voices (Stein, 1979), but far less accurate in the second and third sections where an increasing degree of variation of the original material occurs. Also, in sections 2 and 3, segment boundaries may not fall on whole-quarter-note time points, instead they may be shifted by a small amount, as an effect of the equal division of the sections. This may result in poor discriminatory information contained in segments when using constant-duration segmentation. The approach based on wavelet representation and segmentation is more robust to variation compared to constant-duration segmentation and the unfiltered pitch signal, resulting in similar classification rates for each classifier among all three sections.

A Wilcoxon signed rank test indicated that the concatenation phase did not significantly affect the results of accuracy per segmentation method (CS: $Z = 1.6036$, $p = 0.1088$, ZC: $Z = 0.4472$, $p = 0.6547$, AM: $Z = 1.6036$, $p = 0.1088$) or accuracy per representation type (VR: $Z = 1.4142$, $p = 0.1573$, WR: $Z = 1.6036$, $p = 0.1088$,

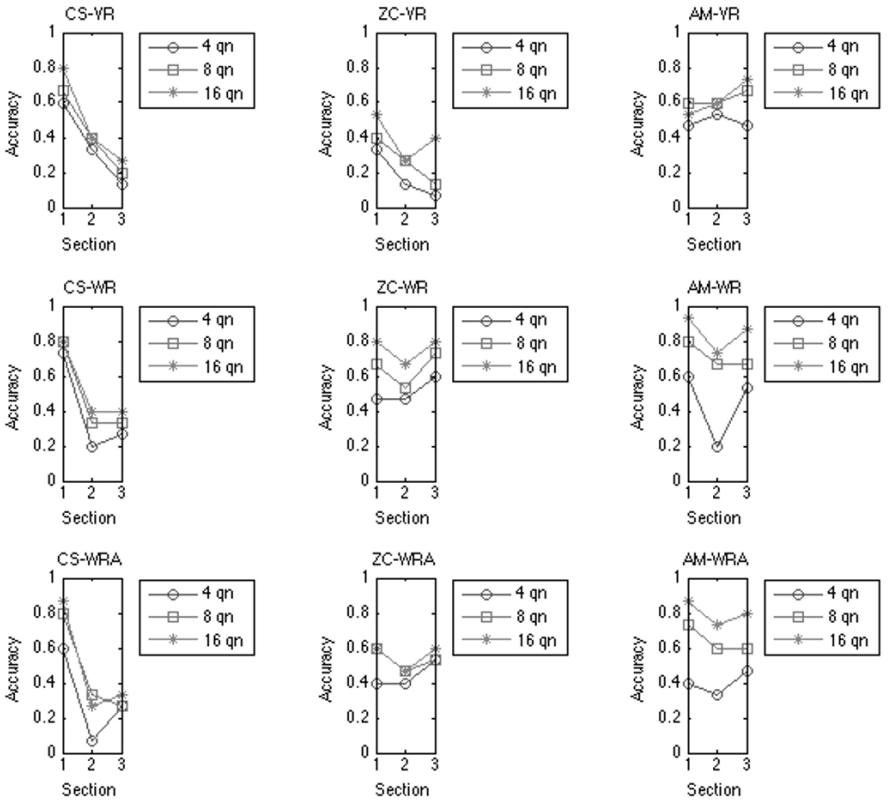

Fig. 12.14 Performance for each section with the classifier based on the exposition

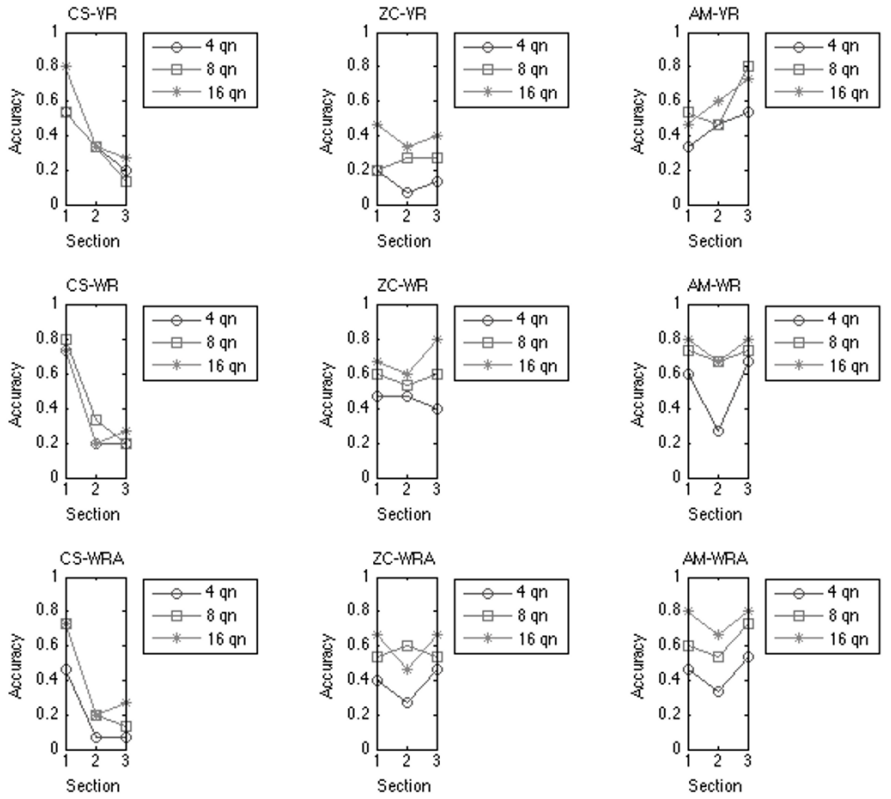

Fig. 12.15 Performance for each section with the classifier based on the exposition, and the concatenation phase included in the segmentation process

WA: $Z = 0.8165$, $p = 0.4142$) for classifiers built from the first 16 qn. However, while including the concatenation phase did not significantly affect the results, it slightly reduced the mean accuracy by 4%. We speculate that this may be a result of the concatenation phase causing some test-set segments to become much longer than the classifier segments, which would lead to segments of very unequal length being measured for similarity. This, in turn, could result in poorer classification accuracies.

12.4 Summary and Conclusions

We have presented a novel computational method for analysis and pattern discovery in melodies and monophonic voices. The method was evaluated on two musicological tasks. In the first task, the method was used to automatically discover themes and sections in the JKU Patterns Development Database. In the second task, the method

was used to determine the parent composition of excerpts from J. S. Bach's Two-Part Inventions (BWV 772–786). We explored aspects of representation, segmentation, classification and ranking of melodic units. The results of the experiments led us to conclude that the combination of constant-duration segmentation and an unfiltered, "raw", pitch-signal representation is a powerful approach for pieces where motivic and thematic material is restated with only slight variation. However, when motivic material is more extensively varied, the wavelet-based approach proves more robust to melodic variation.

The method described in this chapter could be developed further, perhaps by evaluating the quality of clusters in order to discard clusters that are too heterogeneous. Other measures of pattern quality could also be explored for ranking patterns in the algorithm output, including measures that perhaps more precisely model human perception and cognition of musical patterns. Moreover, it would be interesting to study the method's performance on a corpus of human performances of the pieces in experiment 1, in order to test, in particular, the robustness of our distance measures.

Acknowledgements Gissel Velarde is supported by the Department of Architecture, Design and Media Technology at Aalborg University. The contribution of David Meredith to the work reported here was made as part of the "Learning to Create" project (Lrn2Cre8). The project Lrn2Cre8 acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

References

Adiloglu, K., Noll, T., and Obermayer, K. (2006). A paradigmatic approach to extract the melodic structure of a musical piece. *Journal of New Music Research*, 35(3):221–236.

Anagnostopoulou, C. and Westermann, G. (1997). Classification in music: A computational model for paradigmatic analysis. In *Proceedings of the International Computer Music Conference*, pages 125–128, Thessaloniki, Greece.

Antoine, J.-P. (1999). Wavelet analysis: a new tool in physics. In van den Berg, J. C., editor, *Wavelets in Physics*. Cambridge University Press.

Aucouturier, J.-J. and Sandler, M. (2002). Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In *Audio Engineering Society 22nd International Conference on Virtual, Synthetic, and Entertainment Audio (AES22)*, Espoo, Finland.

Cambouropoulos, E. (1997). Musical rhythm: A formal model for determining local boundaries, accents and metre in a melodic surface. In Leman, M., editor, *Music, Gestalt, and Computing*, volume 1317 of *Lecture Notes in Artificial Intelligence*, pages 277–293. Springer.

Cambouropoulos, E. (1998). *Towards a general computational theory of musical structure*. PhD thesis, University of Edinburgh.

Cambouropoulos, E. (2001). The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC'2001)*, Havana, Cuba.

Cambouropoulos, E. and Widmer, G. (2000). Automated motivic analysis via melodic clustering. *Journal of New Music Research*, 29(4):303–317.

Collins, T. (2014). MIREX 2014 Competition: Discovery of Repeated Themes and Sections. http://tinyurl.com/krnqzn5. Accessed on 9 April 2015.

Collins, T., Laney, R., Willis, A., and Garthwaite, P. H. (2011). Modeling pattern importance in Chopin's Mazurkas. *Music Perception*, 28(4):387–414.

Conklin, D. (2006). Melodic analysis with segment classes. *Machine Learning*, 65(2-3):349–360.

Conklin, D. and Anagnostopoulou, C. (2006). Segmental pattern discovery in music. *INFORMS Journal on computing*, 18(3):285–293.

Dreyfus, L. (1996). *Bach and the Patterns of Invention*. Harvard University Press.

Eerola, T. and Toiviainen, P. (2004). MIDI Toolbox: MATLAB tools for music research. Available online at http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox/.

Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley.

Farge, M. (1992). Wavelet transforms and their applications to turbulence. *Annual Review of Fluid Mechanics*, 24(1):395–458.

Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., and Zubrzycki, S. (1951). Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicae*, 2(3–4):282–285.

Forth, J. (2012). *Cognitively-motivated geometric methods of pattern discovery and models of similarity in music*. PhD thesis, Goldsmiths College, University of London.

Grilo, C. F. A., Machado, F., and Cardoso, F. A. B. (2001). Paradigmatic analysis using genetic programming. In *Artificial Intelligence and Simulation of Behaviour (AISB 2001)*, York, UK.

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371.

Höthker, K., Hörnel, D., and Anagnostopoulou, C. (2001). Investigating the influence of representations and algorithms in music classification. *Computers and the Humanities*, 35(1):65–79.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106.

Huron, D. (1996). The melodic arch in Western folksongs. *Computing in Musicology*, 10:3–23.

Janssen, B., De Haas, W. B., Volk, A., and Van Kranenburg, P. (2013). Discovering repeated patterns in music: state of knowledge, challenges, perspectives. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2010)*, Marseille, France.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452(7185):352–355.

Kurby, C. A. and Zacks, J. M. (2008). Segmentation in the perception and memory of events. *Trends in Cognitive Sciences*, 12(2):72–79.

Lamont, A. and Dibben, N. (2001). Motivic structure and the perception of similarity. *Music Perception*, 18(3):245–274.

Lartillot, O. (2005). Efficient extraction of closed motivic patterns in multi-dimensional symbolic representations of music. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pages 191–198, London, UK. Available online at <http://ismir2005.ismir.net/proceedings/1082.pdf>.

Lartillot, O. (2014). PatMinr: In-depth motivic analysis of symbolic monophonic sequences. In *Music Information Retrieval Evaluation Exchange (MIREX 2014), Competition on Discovery of Repeated Themes and Sections*.

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.

Mallat, S. (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd edition.

Mallat, S. and Hwang, W. L. (1992). Singularity detection and processing with wavelets. *Information Theory, IEEE Transactions on*, 38(2):617–643.

Marx, A. B. (1837). *Die Lehre von der musikalischen Komposition: praktisch-theoretisch*, volume 1. Breitkopf and Härtel.

Mazzola, G. et al. (2002). *The Topos of Music*. Birkhäuser.

McGettrick, P. (1997). *MIDIMatch: Musical pattern matching in real time*. PhD thesis, MSc. Dissertation, York University, UK.

Meredith, D. (2006). Point-set algorithms for pattern discovery and pattern matching in music. In *Proceedings of the Dagstuhl Seminar on Content-based Retrieval (No. 06171, 23–28 April, 2006)*, Schloss Dagstuhl, Germany. Available online at http://drops.dagstuhl.de/opus/volltexte/2006/652.

Meredith, D. (2013). COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In *Music Information Retrieval Evaluation Exchange (MIREX)*, Curitiba, Brazil.

Meredith, D. (2015). Music analysis and point-set compression. *Journal of New Music Research*, 44(3). In press.

Meredith, D., Lemström, K., and Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Monelle, R. (1992). *Linguistics and Semiotics in Music*. Harwood Academic.

Müllensiefen, D. and Wiggins, G. (2011a). Polynomial functions as a representation of melodic phrase contour. In Schneider, A. and von Ruschkowski, A., editors, *Systematic Musicology: Empirical and Theoretical Studies*, volume 28 of *Hamburger Jahrbuch für Musikwissenschaft*. Peter Lang.

Müllensiefen, D. and Wiggins, G. A. (2011b). Sloboda and Parker's recall paradigm for melodic memory: a new, computational perspective. In Deliége, I. and Davidson, J. W., editors, *Music and the Mind: Essays in Honour of John Sloboda*, pages 161–188. Oxford University Press.

Müller, M. (2007). *Information Retrieval for Music and Motion*, volume 2. Springer.

Muzy, J., Bacry, E., and Arneodo, A. (1991). Wavelets and multifractal formalism for singular signals: application to turbulence data. *Physical Review Letters*, 67(25):3515.

Nattiez, J.-J. (1975). *Fondements d'une sémiologie de la musique*. Union Générale d'Éditions.

Nattiez, J.-J. (1986). La sémiologie musicale dix ans après. *Analyse musicale*, 2:22–33.

Nieto, O. and Farbood, M. (2013). Mirex 2013: Discovering musical patterns using audio structural segmentation techniques. In *Music Information Retrieval Evaluation eXchange (MIREX 2013)*, Curitiba, Brazil.

Nieto, O. and Farbood, M. M. (2014). Mirex 2014 entry: Music segmentation techniques and greedy path finder algorithm to discover musical patterns. In *Music Information Retrieval Evaluation Exchange (MIREX 2014)*, Taipei, Taiwan.

Pinto, A. (2009). Indexing melodic sequences via wavelet transform. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 882–885. IEEE.

Reicha, A. (1814). *Traité de mélodie*. Chez l'auteur.

Riemann, H. (1912). *Handbuch der Phrasierung*. Hesse.

Ruwet, N. (1966). Méthodes d'analyses en musicologie. *Revue belge de musicologie*, 20(1/4):65–90.

Schenker, H. (1935). *Der freie Satz*. Universal Edition. (Published in English as E. Oster (trans., ed.) *Free Composition*, Longman, New York, 1979.).

Schmuckler, M. A. (1999). Testing models of melodic contour similarity. *Music Perception*, 16(3):295–326.

Schoenberg, A. (1967). *Fundamentals of Musical Composition*. Faber.

Smith, L. M. and Honing, H. (2008). Time–frequency representation of musical rhythm by continuous wavelets. *Journal of Mathematics and Music*, 2(2):81–97.

Sneath, P. H. (1957). The application of computers to taxonomy. *Journal of General Microbiology*, 17(1):201–226.

Stein, L. (1979). *Structure & style: the study and analysis of musical forms*. Summy-Birchard Company.

Tenney, J. and Polansky, L. (1980). Temporal gestalt perception in music. *Journal of Music Theory*, 24(2):205–241.

Torrence, C. and Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78.

Urbano, J. (2013). Mirex 2013 symbolic melodic similarity: A geometric model supported with hybrid sequence alignment. In *Music Information Retrieval Evaluation Exchange (MIREX 2013)*, Curitiba, Brazil.

van Kranenburg, P., Volk, A., and Wiering, F. (2013). A comparison between global and local features for computational classification of folk song melodies. *Journal of New Music Research*, 42(1):1–18.

Velarde, G. and Meredith, D. (2014). A wavelet-based approach to the discovery of themes and sections in monophonic melodies. In *Music Information Retrieval Evaluation Exchange (MIREX 2014)*, Taipei, Taiwan.

Velarde, G., Weyde, T., and Meredith, D. (2013). An approach to melodic segmentation and classification based on filtering with the Haar-wavelet. *Journal of New Music Research*, 42(4):325–345.

Weyde, T. (2001). Grouping, similarity and the recognition of rhythmic structure. In *Proceedings of the International Computer Music Conference (ICMC)*, Havana, Cuba.

Weyde, T. (2002). Integrating segmentation and similarity in melodic analysis. In *Proceedings of the International Conference on Music Perception and Cognition*, pages 240–243, Sydney, Australia.

Woody, N. A. and Brown, S. D. (2007). Selecting wavelet transform scales for multivariate classification. *Journal of Chemometrics*, 21(7-9):357–363.