Pattern induction and matching in music signals

Anssi Klapuri

Centre for Digital Music, Queen Mary University of London
Mile End Road, E1 4NS London, United Kingdom
anssi.klapuri@eecs.qmul.ac.uk
http://www.elec.qmul.ac.uk/people/anssik/

Abstract. This paper discusses techniques for pattern induction and matching in musical audio. At all levels of music - harmony, melody, rhythm, and instrumentation - the temporal sequence of events can be subdivided into shorter patterns that are sometimes repeated and transformed. Methods are described for extracting such patterns from musical audio signals (pattern induction) and computationally feasible methods for retrieving similar patterns from a large database of songs (pattern matching).

1 Introduction

Pattern induction and matching plays an important part in understanding the structure of a given music piece and in detecting similarities between two different music pieces. The term *pattern* is here used to refer to sequential structures that can be characterized by a time series of feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. The vectors $\mathbf{x}_t$ may represent acoustic features calculated at regularly time intervals or discrete symbols with varying durations. Many different elements of music can be represented in this form, including melodies, drum patterns, and chord sequences, for example.

In order to focus on the desired aspect of music, such as the drums track or the lead vocals, it is often necessary to extract that part from a polyphonic music signal. Section 2 of this paper will discuss methods for separating meaningful musical objects from polyphonic recordings.

Contrary to speech, there is no global dictionary of patterns or "words" that would be common to all music pieces, but in a certain sense, the dictionary of patterns is created anew in each music piece. The term *pattern induction* here refers to the process of learning to recognize sequential structures from repeated exposure [1]. Repetition plays an important role here: rhythmic patterns are repeated, melodic phrases recur and vary, and even entire sections, such as the chorus in popular music, are repeated. This kind of self-reference is crucial for imposing structure on a music piece and enables the induction of the underlying prototypical patterns. Pattern induction will be discussed in Sec. 3.

Pattern matching, in turn, consists of searching a database of music for segments that are similar to a given query pattern. Since the target matches can in principle be located at any temporal position and are not necessarily scaled

to the same length as the query pattern, temporal alignment of the query and target patterns poses a significant computational challenge in large databases. Given that the alignment problem can be solved, another pre-requisite for meaningful pattern matching is to define a distance measure between musical patterns of different kinds. These issues will be discussed in Sec. 4.

Pattern processing in music has several interesting applications, including music information retrieval, music classification, cover song identification, and creation of mash-ups by blending matching excerpts from different music pieces. Given a large database of music, quite detailed queries can be made, such as searching for a piece that would work as an accompaniment for a user-created melody.

2 Extracting the object of interest from music

There are various levels at which pattern induction and matching can take place in music. At one extreme, a polyphonic music signal is considered as a coherent whole and features describing its harmonic or timbral aspects, for example, are calculated. In a more analytic approach, some part of the signal, such as the melody or the drums, is extracted before the feature calculation. Both of these approaches are valid from the perceptual viewpoint. Human listeners, especially trained musicians, can switch between a "holistic" listening mode and a more analytic one where they focus on the part played by a particular instrument or decompose music into its constituent elements and their releationships [2, 3].

Even when a music signal is treated as a coherent whole, it is necessary to transform the acoustic waveform into a series of feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ that characterize the desired aspect of the signal. Among the most widely used features are Mel-frequency cepstral coefficients (MFCCs) to represent the timbral content of a signal in terms of its spectral energy distribution [4]. The local harmonic content of a music signal, in turn, is often summarized using a 12-dimensional chroma vector that represents the amount of spectral energy falling at each of the 12 tones of an equally-tempered scale [5, 6]. Rhythmic aspects are conveniently represented by the modulation spectrum which encodes the pattern of sub-band energy fluctuations within windows of approximately one second in length [7, 8]. Besides these, there are a number of other acoustic features, see [9] for an overview.

Focusing pattern extraction on a certain instrument or part in polyphonic music requires that the desired part be pulled apart from the rest before the feature extraction. While this is not entirely straightforward in all cases, it enables musically more interesting pattern induction and matching, such as looking at the melodic contour independently of the accompanying instruments. Some strategies towards decomposing a music signal into its constituent parts are discussed in the following.

2.1 Time-frequency and spatial analysis

Musical sounds, like most natural sounds, tend to be sparse in the time-frequency domain, meaning that the sounds can be approximated using a small number of non-zero elements in the time-frequency domain. This facilitates sound source separation and audio content analysis. Usually the short-time Fourier transform (STFT) is used to represent a given signal in the time-frequency domain. A viable alternative for STFT is the constant-Q transform (CQT), where the center frequencies of the frequency bins are geometrically spaced [10, 11]. CQT is often ideally suited for the analysis of music signals, since the fundamental frequencies (F0s) of the tones in Western music are geometrically spaced.

Spatial information can sometimes be used to organize time-frequency components to their respective sound sources [12]. In the case of stereophonic audio, time-frequency components can be clustered based on the ratio of left-channel amplitude to the right, for example. This simple principle has been demonstrated to be quite effective for some music types, such as jazz [13], despite the fact that overlapping partials partly undermine the idea. Duda et al. [14] used stereo information to extract the lead vocals from complex audio for the purpose of query-by-humming.

2.2 Separating percussive sounds from the harmonic part

It is often desirable to analyze the drum track of music separately from the harmonic part. The sinusoids+noise model is the most widely-used technique for this purpose [15]. It produces quite robust quality for the noise residual, although the sinusoidal (harmonic) part often suffers quality degradation for music with dense sets of sinusoids, such as orchestral music.

Ono et al. proposed a method which decomposes the power spectrogram $\mathbf{X}_{(F\times T)}$ of a mixture signal into a harmonic part $\mathbf{H}$ and percussive part $\mathbf{P}$ so that $\mathbf{X} = \mathbf{H} + \mathbf{P}$ [16]. The decomposition is done by minimizing an objective function that measures variation over time n for the harmonic part and variation over frequency k for the percussive part. The method is straightforward to implement and produces good results.

Non-negative matrix factorization (NMF) is a technique that decomposes the spectrogram of a music signal into a linear sum of components that have a fixed spectrum and time-varying gains [17, 18]. Helen and Virtanen used the NMF to separate the magnitude spectrogram of a music signal into a couple of dozen components and then used a support vector machine (SVM) to classify each component either to pitched instruments or to drums, based on features extracted from the spectrum and the gain function of each component [19].

2.3 Extracting melody and bass line

Vocal melody is usually the main focus of attention for an average music listener, especially in popular music. It tends to be the part that makes music memorable and easily reproducible by singing or humming [20].

Several different methods have been proposed for the main melody extraction from polyphonic music. The task was first considered by Goto [21] and later various methods for melody tracking have been proposed by Paiva et al. [22], Ellis and Poliner [23], Dressler [24], and Ryynänen and Klapuri [25]. Typically, the methods are based on framewise pitch estimation followed by tracking or streaming over time. Some methods involve a timbral model [21, 26, 23] or a musicological model [27]. For comparative evaluations of the different methods, see [28] and [www.music-ir.org/mirex/].

Melody extraction is closely related to vocals separation: extracting the melody faciliatates lead vocals separation, and vice versa. Several different approaches have been proposed for separating the vocals signal from polyphonic music, some based on tracking the pitch of the main melody [29–31], some based on timbre models for the singing voice and for the instrumental background [32, 33], and yet others utilizing stereo information [13, 14].

Bass line is another essential part in many music types and usually contains a great deal of repetition and note patterns that are rhythmically and tonally interesting. Indeed, high-level features extracted from the bass line and the playing style have been successfully used for music genre classification [34]. Methods for extracting the bass line from polyphonic music have been proposed by Goto [21], Hainsworth [35], and Ryynänen [27].

2.4 Instrument separation from polyphonic music

For human listeners, it is natural to organize simultaneously occurring sounds to their respective sound sources. When listening to music, people are often able to focus on a given instrument – despite the fact that music intrinsically tries to make co-occurring sounds "blend" as well as possible.

Separating the signals of individual instruments from a music recording has been recently studied using various approaches. Some are based on grouping sinusoidal components to sources (see e.g. [36]) whereas some others utilize a structured signal model [37, 38]. Some methods are based on supervised learning of instrument-specific harmonic models [39], whereas recently several methods have been proposed based on unsupervised methods [40–42]. Some methods do not aim at separating time-domain signals, but extract the relevant information (such as instrument identities) directly in some other domain [43].

Automatic instrument separation from a monaural or stereophonic recording would enable pattern induction and matching for the individual instruments. However, source separation from polyphonic music is extremely challenging and the existing methods are generally not as reliable as those intended for melody or bass line extraction.

3 Pattern induction

Pattern induction deals with the problem of detecting repeated sequential structures in music and learning the pattern underlying these repetitions. In the

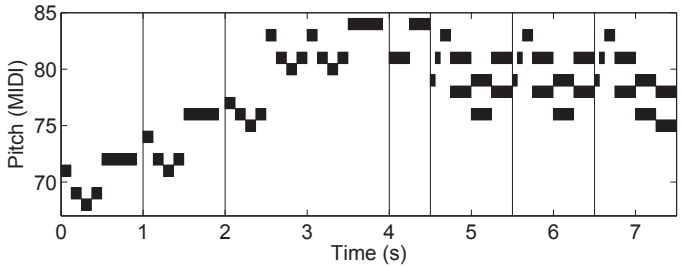

Fig. 1. A "piano-roll" representation for an excerpt from Mozart's *Turkish March*. The vertical lines indicate a possible grouping of the component notes into phrases.

following, we discuss the problem of musical pattern induction from a general perspective. We assume that a time series of feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ describing the desired characteristics of the input signal is given. The task of pattern induction, then, is to detect repeated sequences in this data and to learn a prototypical pattern that can be used to represent all its occurrences. What makes this task challenging is that the data is generally multidimensional and real-valued (as opposed to symbolic data), and furthermore, music seldom repeats itself exactly, but variations and transformations are applied on each occurrence of a given pattern.

3.1 Pattern segmentation and clustering

The basic idea of this approach is to subdivide the feature sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ into shorter segments and then cluster these segments in order to find repeated patterns. The clustering part requires that a distance measure between two feature segments is defined – a question that will be discussed separately in Sec. 4 for different types of features.

For pitch sequences, such as the melody and bass lines, there are well-defined musicological rules how individual sounds are perceptually grouped into melodic phrases and further into larger musical entities in a hierarchical manner [44]. This process is called *grouping* and is based on relatively simple principles such as preferring a phrase boundary at a point where the time or the pitch interval between two consecutive notes is larger than in the immediate vicinity (see Fig. 1 for an example). Pattern induction, then, proceeds by choosing a certain time scale, performing the phrase segmentation, cropping the pitch sequences according to the shortest phrase, clustering the phrases using for example k-means clustering, and finally using the pattern nearest to each cluster centroid as the "prototype" pattern for that cluster.

A difficulty in implementing the phrase segmentation for audio signals is that contrary to MIDI, note durations and rests are difficult to extract from audio. Nevertheless, some methods produce discrete note sequences from music [27, 45], and thus enable segmenting the transcription result into phrases.

Musical *meter* is an alternative criterion for segmenting musical feature sequences into shorter parts for the purpose of clustering. Computational meter analysis usually involves tracking the beat and locating bar lines in music. The good news here is that meter analysis is a well-understood and feasible problem for audio signals too (see e.g. [46]). Furthermore, melodic phrase boundaries often co-incide with strong beats, although this is not always the case. For melodic patterns, for example, this segmenting rule effectively requires two patterns to be similarly positioned with respect to the musical measure boundaries in order for them to be similar, which may sometimes be a too strong assumption. However, for drum patterns this requirement is well justified.

Bertin-Mahieux et al. performed harmonic pattern induction for a large database of music in [47]. They calculated a 12-dimensional chroma vector for each musical beat in the target songs. The beat-synchronous chromagram data was then segmented at barline positions and the resulting beat-chroma patches were vector quantized to obtain a couple of hundred prototype patterns.

A third strategy is to avoid segmentation altogether by using *shift-invariant* features. As an example, let us consider a sequence of one-dimensional features $x_1, x_2, \ldots, x_T$. The sequence is first segmented into partly-overlapping frames that have length approximately the same as the patterns being sought. Then the sequence within each frame is Fourier transformed and the phase information is discarded in order to make the features shift-invariant. The resulting magnitude spectra are then clustered to find repeated patterns. The modulation spectrum features (aka fluctuation patterns) mentioned in the beginning of Sec. 2 are an example of such a shift-invariant feature [7, 8].

3.2 Self-distance matrix

Pattern induction, in the sense defined in the beginning of this section, is possible only if a pattern is repeated in a given feature sequence. The repetitions need not be identical, but bear some similarity with each other. A self-distance matrix (aka self-similarity matrix) offers a direct way of detecting these similarities. Given a feature sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ and a distance function d that specifies the distance between two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the self-distance matrix (SDM) is defined as

$$D(i, j) = d(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

for $i, j \in \{1, 2, \ldots, T\}$. Frequently used distance measures include the Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ and the cosine distance $0.5(1 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle / (\|\mathbf{x}_i\| \|\mathbf{x}_j\|))$. Repeated sequences appear in the SDM as off-diagonal stripes. Methods for detecting these will be discussed below.

An obvious difficulty in calculating the SDM is that when the length T of the feature sequence is large, the number of distance computations T^2 may become computationally prohibitive. A typical solution to overcome this is to use beat-synchronized features: a beat tracking algorithm is applied and the features $\mathbf{x}_t$ are then calculated within (or averaged over) each inter-beat interval. Since the average inter-beat interval is approximately 0.5 seconds – much larger than a

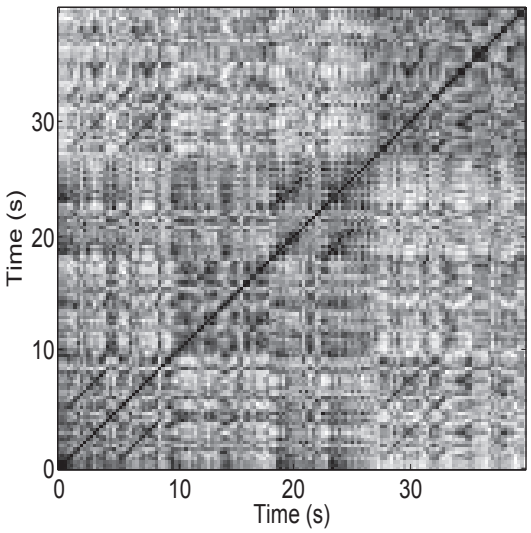

Fig. 2. A self-distance matrix for Chopin's *Etude Op 25 No 9*, calculated using beat-synchronous chroma features. As the off-diagonal dark stripes indicate, the note sequence between 1s and 5s starts again at 5s, and later at 28s and 32s in a varied form.

typical analysis frame size – this greatly reduces the number of elements in the time sequence and in the SDM. An added benefit of using beat-synchronous features is that this compensates for tempo fluctuations within the piece under analysis. As a result, repeated sequences appear in the SDM as stripes that run exactly parallel to the main diagonal. Figure 2 shows an example SDM calculated using beat-synchronous chroma features.

Self-distance matrices have been widely used for audio-based analysis of the sectional form (structure) of music pieces [48, 49]. In that domain, several different methods have been proposed for localizing the off-diagonal stripes that indicate repeating sequences in the music [50–52]. Goto, for example, first calculates a marginal histogram which indicates the diagonal bands that contain considerable repetition, and then finds the beginning and end points of the repeted segments at a second step [51]. Serra has proposed an interesting method for detecting locally similar sections in two feature sequences [53].

3.3 Lempel-Ziv-Welch family of algorithms

Repeated patterns are heavily utilized in universal lossless data compression algorithms. The Lempel-Ziv-Welch (LZW) algorithm, in particular, is based on matching and replacing repeated patterns with code values [54]. Let us denote a sequence of discrete symbols by $s_1, s_2, \ldots, s_T$. The algorithm initializes a dictionary which contains codes for individual symbols that are possible at the input. At the compression stage, the input symbols are gathered into a sequence until

the next character would make a sequence for which there is no code yet in the dictionary, and a new code for that sequence is then added to the dictionary.

The usefulness of the LZW algorithm for musical pattern matching is limited by the fact that it requires a sequence of *discrete symbols* as input, as opposed to real-valued feature vectors. This means that a given feature vectore sequence has to be vector-quantized before processing with the LZW. In practice, also beat-synchronous feature extraction is needed to ensure that the lengths of repeated sequences are not affected by tempo fluctuation. Vector quantization (VQ, [55]) as such is not a problem, but choosing a suitable level of granularity becomes very difficult: if the number of symbols is too large, then two repeats of a certain pattern are quantized dissimilarly, and if the number of symbols is too small, too much information is lost in the quantization and spurious repeats are detected.

Another inherent limitation of the LZW family of algorithms is that they require *exact* repetition. This is usually not appropriate in music, where variation is more a rule than an exception. Moreover, the beginning and end times of the learned patterns are arbitrarily determined by the order in which the input sequence is analyzed. Improvements over the LZW family of algorithms for musical pattern induction have been considered e.g. by Lartillot et al. [56].

3.4 Markov models for sequence prediction

Pattern induction is often used for the purpose of *predicting* a data sequence. N-gram models are a popular choice for predicting a sequence of discrete symbols $s_1, s_2, \ldots, s_T$ [57]. In an N-gram, the preceding $N-1$ symbols are used to determine the probabilities for different symbols to appear next, $P(s_t|s_{t-1}, \ldots, s_{t-N+1})$. Increasing N gives more accurate predictions, but requires a very large amount of training data to estimate the probabilities reliably. A better solution is to use a variable-order Markov model (VMM) for which the context length N varies in response to the available statistics in the training data [58]. This is a very desirable feature, and for note sequences, this means that both short and long note sequences can be modeled within a single model, based on their occurrences in the training data. Probabilistic predictions can be made even when patterns do not repeat exactly.

Ryynänen and Klapuri used VMMs as a predictive model in a method that transcribes bass lines in polyphonic music [59]. They used the VMM toolbox of Begleiter et al. for VMM training and prediction [58].

3.5 Interaction between pattern induction and source separation

Music often introduces a certain pattern to the listener in a simpler form before adding further "layers" of instrumentation at subsequent repetitions (and variations) of the pattern. Provided that the repetitions are detected via pattern induction, this information can be fed back in order to improve the separation and analysis of certain instruments or parts in the mixture signal. This idea was used by Mauch et al. who used information about music structure to improve recognition of chords in music [60].

4 Pattern matching

This section considers the problem of searching a database of music for segments that are similar to a given pattern. The query pattern is denoted by a feature seqence $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M$, and for convenience, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ is used to denote a concatenation of the feature sequences extracted from all target music pieces.

Before discussing the similarity metrics between two music patterns, let us consider the general computational challenges in comparing a query pattern against a large database, an issue that is common to all types of musical patterns.

4.1 Temporal alignment problem in pattern comparison

Pattern matching in music is computationally demanding, because the query pattern can in principle occur at any position of the target data and because the time-scale of the query pattern may differ from the potential matches in the target data due to tempo differences. These two issues are here referred to as the time-shift and time-scale problem, respectively. Brute-force matching of the query pattern at all possible locations of the target data and using different time-scaled versions of the query pattern would be computationally infeasible for any database of significant size.

Dynamic time warping (DTW) is a technique that aims at solving both the time-shift and time-scale problem simultaneously. In DTW, a matrix of distances is computed so that element (i, j) of the matrix represents the pair-wise distance between element i of the query pattern and element j in the target data (see Fig. 3 for an example). Dynamic programming is then applied to find a path of small distances from the first to the last row of the matrix, placing suitable constraints on the geometry of the path. DTW has been used for melodic pattern matching by Dannenberg [61], for structure analysis by Paulus [52], and for cover song detection by Serra [53], to mention a few examples.

Beat-synchronous feature extraction is an efficient mechanism for dealing with the time-scale problem, as already discussed in Sec. 3. To allow some further flexibility in pattern scaling and to mitigate the effect of tempo estimation errors, it is sometimes useful to further time-scale the beat-synchronized query pattern by factors $\frac{1}{2}$, 1, and 2, and match each of these separately.

A remaining problem to be solved is the temporal shift: if the target database is very large, comparing the query pattern at every possible temporal position in the database can be infeasible. Shift-invariant features are one way of dealing with this problem: they can be used for approximate pattern matching to prune the target data, after which the temporal alignment is computed for the best-matching candidates. This allows the first stage of matching to be performed an order of magnitude faster.

Another potential solution for the time-shift problem is to segment the target database by meter analysis or grouping analysis, and then match the query pattern only at temporal positions determined by estimated bar lines or group boundaries. This approach was already discussed in Sec. 3.

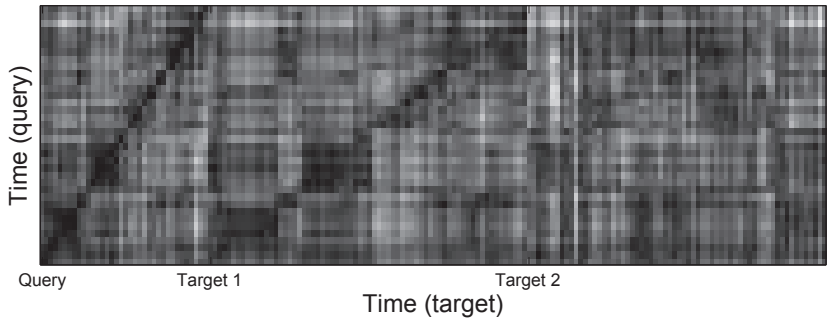

Fig. 3. A matrix of distances used by DTW to find a time-alignment between different feature sequences. The vertical axis represents the time in a query excerpt (Queen's *Bohemian Rhapsody*). The horizontal axis corresponds to the concatenation of features from three different excerps: 1) the query itself, 2) "Target 1" (*Bohemian Rhapsody* performed by London Symphonium Orchestra) and Target 2 (*It's a Kind of Magic* by Queen). Beginnings of the three targets are indicated below the matrix. Darker values indicate smaller distance.

Finally, efficient indexing techniques exist for dealing with extremely large databases. In practice, these require that the time-scale problem is eliminated (e.g. using beat-synchronous features) and the number of time-shifts is greatly reduced (e.g. using shift-invariant features or pre-segmentation). If these conditions are satisfied, the locality sensitive hashing (LSH) for example, enables sublinear search complexity for retrieving the approximate nearest neighbours of the query pattern from a large database [62]. Ryynanen et al. used LSH for melodic pattern matching in [63].

4.2 Melodic pattern matching

Melodic pattern matching is usually considered in the context of query-by-humming (QBH), where a user's singing or humming is used as a query to retrieve music with a matching melodic fragment. Typically, the user's singing is first transcribed into a pitch trajectory or a note sequence before the matching takes place. QBH has been studied for more than 15 years and remains an active research topic [64, 65].

Research on QBH originated in the context of the retrieval from MIDI or score databases. Matching approaches include string matching techniques [66], hidden Markov models [67, 68], dynamic programming [69, 70], and efficient recursive alignment [71]. A number of QBH systems have been evaluated in Music Information Retrieval Evaluation eXchange (MIREX) [72].

Methods for the QBH of audio data have been proposed only quite recently [73, 74, 14, 75, 63]. Typically, the methods extract the main melodies from the target musical audio (see Sec. 2.3) before the matching takes place. However, it should be noted that a given query melody can in principle be matched di-

rectly against polyphonic audio data in the time-frequency or time-pitch domain. Some on-line services incorporating QBH are already available, see e.g. [www.midomi.com], [www.musicline.de], [www.musipedia.org].

Matching two melodic patterns requires a proper definition of similarity. The trivial assumption that two patterns are similar if they have identical pitches is usually not appropriate. There are three main reasons that cause the query pattern and the target matches to differ: 1) low quality of the sung queries (especially in the case of musically untrained users), 2) errors in extracting the main melodies automatically from music recordings, and 3) musical variation, such as fragmentation (elaboration) or consolidation (reduction) of a given melody [44]. One approach that works quite robustly in the presence of all these factors is to calculate Euclidean distance between temporally aligned log-pitch trajectories. Musical key normalization can be implemented simply by normalizing the two pitch contours to zero mean. More extensive review of research on melodic similarity can be found in [76].

4.3 Patterns in polyphonic pitch data

Instead of using only the main melody for music retrieval, polyphonic pitch data can be processed directly. Multipitch estimation algorithms (see [77, 78] for review) can be used to extract multiple pitch values in successive time frames, or alternatively, a mapping from time-frequency to a time-pitch representation can be employed [79]. Both of these approaches yield a representation in the time-pitch plane, the difference being that multipitch estimation algorithms yield a discrete set of pitch values, whereas mapping to a time-pitch plane yields a more continuous representation. Matching a query pattern against a database of music signals can be carried out by a two-dimensional correlation analysis in the time-pitch plane.

4.4 Chord sequences

Here we assume that chord information is represented as a discrete symbol sequence $s_1, s_2, \ldots, s_T$, where s_t indicates the chord identity at time frame t. Measuring the distance between two chord sequences requires that the distance between each pair of different chords is defined. Often this distance is approximated by arranging chords in a one- or two-dimensional space, and then using the geometric distance between chords in this space as the distance measure [80], see Fig. 4 for an example. In the one-dimensional case, the circle of fifths is often used.

It is often useful to compare two chord sequences in a key-invariant manner. This can be done by expressing chords in relation to tonic (that is, using chord degrees instead of the "absolute" chords), or by comparing all the 12 possible transformations and choosing the minimum distance.

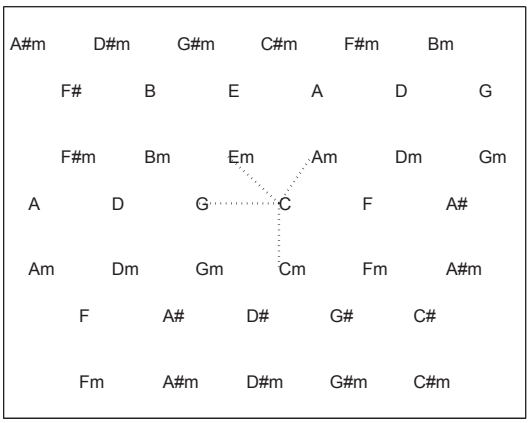

Fig. 4. Major and minor triads arranged in a two dimensional chord space. Here the Euclidean distance between each two points can be used to approximate the distance between chords. The dotted lines indicate the four distance parameters that define this particular space.

4.5 Drum patterns and rhythms

Here we discuss pattern matching in drum tracks that are presented as acoustic signals and are possibly extracted from polyphonic music using the methods described in Sec. 2.2. Applications of this include for example query-by-tapping [www.music-ir.org/mirex/] and music retrieval based on drum track similarity.

Percussive music devoid of both harmony and melody can contain considerable amount of musical form and structure, encoded into the timbre, loudness, and timing relationships between the component sounds. Timbre and loudness characteristics can be conveniently represented by MFCCs extracted in successive time frames. Often, however, the absolute spectral shape and loudness of the components sounds is not of interest, but instead, the timbre and loudness of sounds relative to each other defines the perceived rhythm. Paulus and Klapuri reduced the rhythmic information into a two-dimensional signal describing the evolution of loudness and spectral centroid over time, in order to compare rhythmic patterns performed using an arbitrary set of sounds [81]. The features were mean- and variance-normalized to allow comparison across different sound sets, and DTW was used to align the two patterns under comparison.

Ellis and Arroyo projected drum patterns into a low-dimensional representation, where different rhythms could be represented as a linear sum of so-called eigenrhythms [82]. They collected 100 drum patterns from popular music tracks and estimated the bar line positions in these. Each pattern was normalized and the resulting set of patterns was subjected to principal component analysis in order to obtain a set of basis patterns ("eigenrhythms") that were then combined to approximate the original data. The low-dimensional representation of the drum patterns was used as a space for classification and for measuring similarity between rhythms.

Non-negative matrix factorization (NMF, see Sec. 2.2) is another technique for obtaining a mid-level representation for drum patterns [83]. The resulting component gain functions can be subjected to the eigenrhythm analysis described above, or statistical measures can be calculated to characterize the spectra and gain functions for rhythm comparison.

5 Conclusions

This paper has discussed the induction and matching of sequential patterns in musical audio. Such patterns are neglected by the commonly used "bag-of-features" approach to music retrieval, where statistics over feature vectors are calculated to collapse the time structure altogether. Processing sequentical structures poses computational challenges, but also enables musically interesting retrieval tasks beyond those possible with the bag-of-features approach. Some of these applications, such as query-by-humming services, are already available for consumers.

Acknowledgments. Thanks to Jouni Paulus for the Matlab code for computing self-distance matrices. Thanks to Christian Dittmar for the idea of using repeated patterns to improve the accuracy of source separation and analysis.

References

1. R. Rowe, *Machine musicianship*. Cambridge, Massachusetts: MIT Press, 2001.
2. T. G. Bever and R. J. Chiarello, "Cerebral dominance in musicians and nonmusicians," *The Journal of Neuropsychiatry and Clinical Neurosciences*, vol. 21, no. 1, pp. 94–97, 2009.
3. J. Barbour, "Analytic listening: A case study of radio production," in *International Conference on Auditory Display*, Sydney, Australia, July 2004.
4. K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis – a unified approach to speech spectral estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, 1994.
5. M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2001, pp. 15–18.
6. M. Müller, S. Ewert, and S. Kreuzer, "Making chroma features more robust to timbre changes," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 1869–1872.
7. S. Dixon, E. Pampalk, and G. Widmer, "Classification of dance music by periodicity patterns," in *4th International Conference on Music Information Retrieval*, Baltimore MD, 2003, pp. 159–165.
8. K. Jensen, "Multiple scale music segmentation using rhythm, timbre, and harmony," *EURASIP Journal on Advances in Signal Processing*, 2007.
9. G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," IRCAM, Paris, France, Tech. Rep., Apr. 2004.

10. J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Am.*, vol. 89, no. 1, pp. 425–434, 1991.
11. C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *7th Sound and Music Computing Conference*, Barcelona, Spain, 2010.
12. O. Yilmaz and S. Richard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
13. D. Barry, B. Lawlor, and E. Coyle, "Sound source separation: Azimuth discrimination and resynthesis," in *7th International Conference on Digital Audio Effects*, Naples, Italy, October 2004, pp. 240–244.
14. A. Duda, A. Nürnberger, and S. Stober, "Towards query by humming/singing on audio databases," in *International Conference on Music Information Retrieval*, Vienna, Austria, 2007, pp. 331–334.
15. X. Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, C. Roads, S. Pope, A. Picialli, and G. D. Poli, Eds. Swets & Zeitlinger, 1997.
16. N. Ono, K. Miyamoto, J. L. Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffucion on spectrogram," in *European Signal Processing Conference*, Lausanne, Switzerland, August 2008, pp. 240–244.
17. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.
18. T. Virtanen, "Unsupervised learning methods for source separation in monaural music signals," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer, 2006, pp. 267–296.
19. M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negtive matrix factorization and support vector machine," in *European Signal Processing Conference*, Antalya, Turkey, 2005.
20. E. Selfridge-Field, "Conceptual and representational issues in melodic comparison," *Computing in Musicology*, vol. 11, pp. 3–64, 1998.
21. M. Goto, "A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.
22. R. P. Paiva, T. Mendes, and A. Cardoso, "On the detection of melody notes in polyphonic audio," in *6th International Conference on Music Information Retrieval*, London, UK, pp. 175–182.
23. D. P. W. Ellis and G. Poliner, "Classification-based melody transcription," *Machine Learning*, vol. 65, no. 2–3, pp. 439–456, 2006.
24. K. Dressler, "An auditory streaming approach on melody extraction," in *Intl. Conf. on Music Information Retrieval*, Victoria, Canada, 2006, MIREX evaluation.
25. M. Ryynänen and A. Klapuri, "Transcription of the singing melody in polyphonic music," in *Intl. Conf. on Music Information Retrieval*, Victoria, Canada, 2006, pp. 222–227.
26. M. Marolt, "Audio melody extraction based on timbral similarity of melodic fragments," in *EUROCON 2005*, November 2005.
27. M. Ryynänen and A. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.
28. G. Poliner, D. Ellis, A. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, May 2007.

29. H. Fujihara and M. Goto, "A music information retrieval system based on singing voice timbre," in *Intl. Conf. on Music Information Retrieval*, Vienna, Austria, 2007.
30. Y. Li and D. L. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1475–1487, May 2007.
31. T. Virtanen, A. Mesaros, and M. Ryynänen, "Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition*, Brisbane, Australia, September 2008.
32. A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1564–1578, 2007.
33. J.-L. Durrieu, G. Richard, B. David, and C. Fevotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 564–575, 2010.
34. J. Abesser, H. Lukashevich, C. Dittmar, and G. Schuller, "Genre classification using bass-related high-level features and playing styles," in *Intl. Society on Music Information Retrieval Conference*, Kobe, Japan, 2009.
35. S. W. Hainsworth and M. D. Macleod, "Automatic bass line transcription from polyphonic music," in *International Computer Music Conference*, Havana, Cuba, 2001, pp. 431–434.
36. J. Burred, A. Röbel, and T. Sikora, "Dynamic spectral envelope modeling for the analysis of musical instrument sounds," *IEEE Trans. Audio, Speech, and Language Processing*, 2009.
37. J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David, "Main instrument separation from stereophonic audio signals using a source/filter model," in *Proc. EUSIPCO*, Glasgow, Scotland, August 2009.
38. R. Badeau, V. Emiya, and B. David, "Expectation-maximization algorithm for multi-pitch estimation and separation of overlapping harmonic spectra," in *Proc. IEEE ICASSP*, Taipei, Taiwan, 2009, pp. 3073–3076.
39. P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 116–128, 2008.
40. D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tenson factorisation models for musical source separation," *Computational Intelligence and Neuroscience*, 2008.
41. E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *IEEE ICASSP*, Las Vegas, USA, 2008.
42. T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.
43. T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrogram: Probabilistic representation of instrument existence for polyphonic music," *IPSJ Journal*, vol. 48, no. 1, pp. 214–226, 2007.
44. F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music.* MIT Press, 1983.
45. C. Yeh, "Multiple fundamental frequency estimation of polyphonic recordings," Ph.D. dissertation, University of Paris VI, 2008.

46. A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Speech and Audio Processing*, vol. 14, no. 1, 2006.
47. T. Bertin-Mahieux, R. J. Weiss, and D. P. W. Ellis, "Clustering beat-chroma patterns in a large music database," in *Proc. of the Int. Society for Music Information Retrieval Conference*, Utrecht, Netherlands, 2010.
48. 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. Springer, 2009, pp. 305–331.
49. J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. of the Int. Society for Music Information Retrieval Conference*, Utrecht, Netherlands, 2010.
50. G. Peeters, "Sequence representations of music structure using higher-order similarity matrix and maximum-likelihood approach," in *Intl. Conf. on Music Information Retrieval*, Vienna, Austria, 2007, pp. 35–40.
51. M. Goto, "A chorus-section detecting method for musical audio signals," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, Hong Kong, China, Apr. 2003, pp. 437–440.
52. J. Paulus, "Signal processing methods for drum transcription and music structure analysis," Ph.D. dissertation, Tampere University of Technology, 2009.
53. J. Serra, E. Gomez, P. Herrera, , and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 1138–1152, 2007.
54. T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, 1984.
55. A. Gersho and R. Gray, *Vector quantization and signal compression.* Kluwer Academic Publishers, 1991.
56. O. Lartillot, S. Dubnov, G. Assayag, and G. Bejerano, "Automatic modeling of musical style," in *International Computer Music Conference*, 2001.
57. D. Jurafsky and J. H. Martin, *Speech and language processing.* New Jersey, USA: Prentice Hall, 2000.
58. R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order Markov models," *J. of Artificial Intelligence Research*, vol. 22, pp. 385–421, 2004.
59. M. Ryynänen and A. Klapuri, "Automatic bass line transcription from streaming polyphonic audio," in *IEEE International Conference on Audio, Speech and Signal Processing*, 2007, pp. 1437–1440.
60. M. Mauch, K. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription," in *Proc. 10th Intl. Society for Music Information Retrieval Conference*, Kobe, Japan, 2009.
61. R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163, 2003.
62. M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *ACM Symposium on Computational Geometry*, 2004, pp. 253–262.
63. M. Ryynänen and A. Klapuri, "Query by humming of MIDI and audio using locality sensitive hashing," in *IEEE International Conference on Audio, Speech and Signal Processing*, Las Vegas, USA, pp. 2249–2252.
64. A. Ghias, J. Logan, and D. Chamberlin, "Query by humming: Musical information retrieval in an audio database," in *ACM Multimedia Conference '95.* San Fransisco, California: Cornell University, Nov. 1995.

65. R. McNab, L. Smith, I. Witten, C. Henderson, and S. Cunningham, "Towards the digital music library: Tune retrieval from acoustic input," in *First ACM International Conference on Digital Libraries*, 1996, pp. 11–18.
66. K. Lemström, "String matching techniques for music retrieval," Ph.D. dissertation, University of Helsinki, 2000.
67. C. Meek and W. Birmingham, "Applications of binary classification and adaptive boosting to the query-by-humming problem," in *Intl. Conf. on Music Information Retrieval*, Paris, France, 2002.
68. J.-S. R. Jang, C.-L. Hsu, and H.-R. Lee, "Continuous HMM and its enhancement for singing/humming query retrieval," in *6th International Conference on Music Information Retrieval*, London, UK, 2005.
69. J.-S. R. Jang and M.-Y. Gao, "A query-by-singing system based on dynamic programming," in *International Workshop on Intelligent Systems Resolutions*, 2000.
70. L. Wang, S. Huang, S. Hu, J. Liang, and B. Xu, "An effective and efficient method for query by humming system based on multi-similarity measurement fusion," in *International Conference on Audio, Language and Image Processing*, July 2008, pp. 471–475.
71. X. Wu, M. Li, J. Yang, and Y. Yan, "A top-down approach to melody match in pitch countour for query by humming," in *International Conference of Chinese Spoken Language Processing*, 2006.
72. J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.
73. T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka, "Music signal spotting retrieval by a humming query using start frame feature dependent continuous dynamic programming," in *2nd Annual International Symposium on Music Information Retrieval*, Bloomington, Indiana, USA, Oct. 2001, pp. 211–218.
74. J. Song, S. Y. Bae, and K. Yoon, "Mid-level music melody representation of polyphonic audio for query-by-humming system," in *Intl. Conf. on Music Information Retrieval*, Paris, France, Oct. 2002, pp. 133–139.
75. L. Guo, X. He, Y. Zhang, and Y. Lu, "Content-based retrieval of polyphonic music objects using pitch contour," in *IEEE International Conference on Audio, Speech and Signal Processing*, Las Vegas, USA, 2008, pp. 2205–2208.
76. R. Typke, "Music retrieval based on melodic similarity," Ph.D. dissertation, Universiteit Utrecht, 2007.
77. A. de Cheveigné, "Multiple F0 estimation," in *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, D. Wang and G. J. Brown, Eds. Wiley–IEEE Press, 2006.
78. A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription.* New York: Springer, 2006.
79. A. Klapuri, "A method for visualizing the pitch content of polyphonic music signals," in *Intl. Society on Music Information Retrieval Conference*, Kobe, Japan, 2009.
80. H. Purwins, "Profiles of pitch classes – circularity of relative pitch and key: Experiments, models, music analysis, and perspectives," Ph.D. dissertation, Berlin University of Technology, 2005.
81. J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," in *Intl. Conf. on Music Information Retrieval*, Paris, France, 2002.
82. D. Ellis and J. Arroyo, "Eigenrhythms: Drum pattern basis sets for classification and generation," in *International Conference on Music Information Retrieval*, Barcelona, Spain.

83. J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," in *European Signal Processing Conference*, Antalya, Turkey, Sep. 2005.