Audio Content-Based Music Retrieval

Peter Grosche[*1], Meinard Müller[*1], and Joan Serrà[†2]

1 **Saarland University and MPI Informatik**
Campus E1-4, 66123 Saarbrücken, Germany
pgrosche@mpi-inf.mpg.de, meinard@mpi-inf.mpg.de

2 **Artificial Intelligence Research Institute (IIIA-CSIC)**
Campus UAB s/n, 08193 Bellaterra, Barcelona, Spain
jserra@iiia.csic.es

Abstract

The rapidly growing corpus of digital audio material requires novel retrieval strategies for exploring large music collections. Traditional retrieval strategies rely on metadata that describe the actual audio content in words. In the case that such textual descriptions are not available, one requires content-based retrieval strategies which only utilize the raw audio material. In this contribution, we discuss content-based retrieval strategies that follow the query-by-example paradigm: given an audio query, the task is to retrieve all documents that are somehow similar or related to the query from a music collection. Such strategies can be loosely classified according to their *specificity*, which refers to the degree of similarity between the query and the database documents. Here, high specificity refers to a strict notion of similarity, whereas low specificity to a rather vague one. Furthermore, we introduce a second classification principle based on *granularity*, where one distinguishes between fragment-level and document-level retrieval. Using a classification scheme based on specificity and granularity, we identify various classes of retrieval scenarios, which comprise *audio identification*, *audio matching*, and *version identification*. For these three important classes, we give an overview of representative state-of-the-art approaches, which also illustrate the sometimes subtle but crucial differences between the retrieval scenarios. Finally, we give an outlook on a user-oriented retrieval system, which combines the various retrieval strategies in a unified framework.

1998 ACM Subject Classification H.5.5 Sound and Music Computing, J.5 Arts and Humanities–Music, H.5.1 Multimedia Information Systems, I.5 Pattern Recognition

Keywords and phrases music retrieval, content-based, query-by-example, audio identification, audio matching, cover song identification

Digital Object Identifier 10.4230/DFU.Vol3.11041.157

1 Introduction

The way music is stored, accessed, distributed, and consumed underwent a radical change in the last decades. Nowadays, large collections containing millions of digital music documents are accessible from anywhere around the world. Such a tremendous amount of readily available music requires retrieval strategies that allow users to explore large music collections in a convenient and enjoyable way. Most audio search engines rely on metadata and textual

* The authors are funded by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI). Meinard Müller is now with Bonn University, Department of Computer Science III, Germany.
† Funded by Consejo Superior de Investigaciones Científicas (JAEDOC069/2010) and Generalitat de Catalunya (2009-SGR-1434).

Multimodal Music Processing. *Dagstuhl Follow-Ups, Vol. 3.* ISBN 978-3-939897-37-8.
Editors: Meinard Müller, Masataka Goto, and Markus Schedl; pp. 157–174

Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

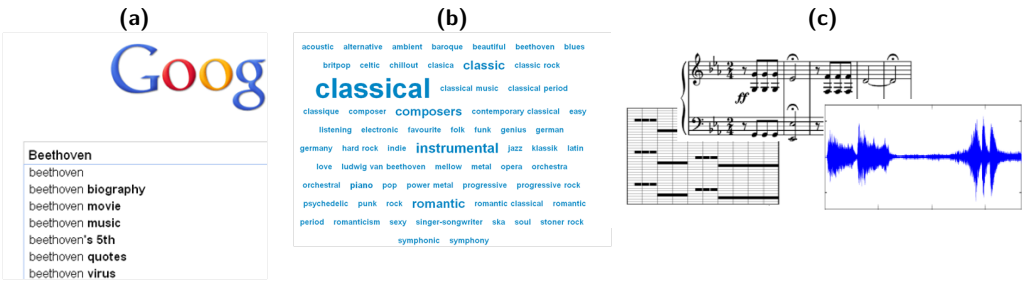

Figure 1 Illustration of retrieval concepts. **(a)** Traditional retrieval using textual metadata (e. g., artist, title) and a web search engine.[1] **(b)** Retrieval based on rich and expressive metadata given by tags.[2] **(c)** Content-based retrieval using audio, MIDI, or score information.

annotations of the actual audio content [11]. Editorial metadata typically include descriptions of the artist, title, or other release information. The drawback of a retrieval solely based on editorial metadata is that the user needs to have a relatively clear idea of what he or she is looking for. Typical query terms may be a title such as "Act naturally" when searching the song by The Beatles or a composer's name such as "Beethoven" (see Figure 1a). In other words, traditional editorial metadata only allow to search for already known content. To overcome these limitations, editorial metadata has been more and more complemented by general and expressive annotations (so called *tags*) of the actual musical content [5, 25, 49]. Typically, tags give descriptions of the musical style or genre of a recording, but may also include information about the mood, the musical key, or the tempo [31, 48]. In particular, tags form the basis for music recommendation and navigation systems that make the audio content accessible even when users are not looking for a specific song or artist but for music that exhibits certain musical properties [49]. The generation of such annotations of audio content, however, is typically a labor intensive and time-consuming process [11, 48]. Furthermore, often musical expert knowledge is required for creating reliable, consistent, and musically meaningful annotations. To avoid this tedious process, recent attempts aim at substituting expert-generated tags by user-generated tags [48]. However, such tags tend to be less accurate, subjective, and rather noisy. In other words, they exhibit a high degree of variability between users. Crowd (or social) tagging, one popular strategy in this context, employs voting and filtering strategies based on large social networks of users for "cleaning" the tags [31]. Relying on the "wisdom of the crowd" rather than the "power of the few" [27], tags assigned by many users are considered more reliable than tags assigned by only a few users. Figure 1b shows the Last.fm[2] *tag cloud* for "Beethoven". Here, the font size reflects the frequency of the individual tags. One major drawback of this approach is that it relies on a large crowd of users for creating reliable annotations [31]. While mainstream pop/rock music is typically covered by such annotations, less popular genres are often scarcely tagged. This phenomenon is also known as the "long-tail" problem [12, 48]. To overcome these problems, *content-based retrieval* strategies have great potential as they do not rely on any manually created metadata but are exclusively based on the audio content and cover the entire audio material in an objective and reproducible way [11]. One possible approach is to employ automated procedures for tagging music, such as automatic genre recognition, mood recognition, or tempo estimation [4, 49]. The major drawback of these learning-based

[1] `www.google.com` (accessed Dec. 18, 2011)
[2] `www.last.fm` (accessed Dec. 18, 2011)

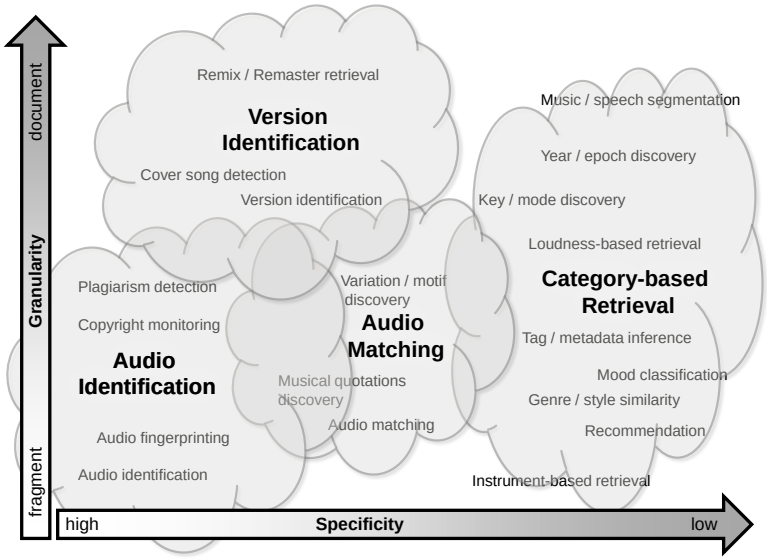

Figure 2 Specificity/granularity pane showing the various facets of content-based music retrieval.

strategies is the requirement of large corpora of tagged music examples as training material and the limitation to queries in textual form. Furthermore, the quality of the tags generated by state-of-the-art procedures does not reach the quality of human generated tags [49].

In this contribution, we present and discuss various retrieval strategies based on audio content that follow the query-by-example paradigm: given an audio recording or a fragment of it (used as query or example), the task is to automatically retrieve documents from a given music collection containing parts or aspects that are similar to it. As a result, retrieval systems following this paradigm do not require any textual descriptions. However, the notion of similarity used to compare different audio recordings (or fragments) is of crucial importance and largely depends on the respective application as well as the user requirements.

Many different audio content-based retrieval systems have been proposed, following different strategies and aiming at different application scenarios. Generally, such retrieval systems can be characterized by various aspects such as the notion of similarity, the underlying matching principles, or the query format. Following and extending the concept introduced in [11], we consider the following two aspects: *specificity* and *granularity*, see Figure 2. The *specificity* of a retrieval system refers to the degree of similarity between the query and the database documents to be retrieved. High-specific retrieval systems return exact copies of the query (in other words, they *identify* the query or occurrences of the query within database documents), whereas low-specific retrieval systems return vague matches that are similar with respect to some musical properties. As in [11], different content-based music retrieval scenarios can be arranged along a specificity axis as shown in Figure 2 (horizontally). We extend this classification scheme by introducing a second aspect, the *granularity* (or temporal scope) of a retrieval scenario. In *fragment-level* retrieval scenarios, the query consists of a short fragment of an audio recording, and the goal is to retrieve all musically related fragments that are contained in the documents of a given music collection. Typically, such fragments may cover only a few seconds of audio content or may correspond to a motif, a theme, or a musical part of a recording. In contrast, in *document-level* retrieval, the query reflects characteristics of an entire document and is compared with entire documents of the database.

Here, the notion of similarity typically is rather coarse and the used features capture global statistics of an entire recording. In this context, one has to distinguish between some kind of internal and some kind of external granularity of the retrieval tasks. In our classification scheme, we use the term fragment-level when a fragment-based similarity measure is used to compare fragments of audio recordings (internal), even though entire documents are returned as matches (external). Using such a classification allows for extending the specificity axis to a specificity/granularity pane as shown in Figure 2. In particular, we have identified four different groups of retrieval scenarios corresponding to the four clouds in Figure 2. Each of the clouds, in turn, encloses a number of different retrieval scenarios. Obviously, the clouds are not strictly separated but blend into each other. Even though this taxonomy is rather vague and sometimes questionable, it gives an intuitive overview of the various retrieval paradigms while illustrating their subtle but crucial differences.

An example of a high-specific fragment-level retrieval task is *audio identification* (sometimes also referred to as *audio fingerprinting* [8]). Given a small audio fragment as query, the task of audio identification consists in identifying the particular audio recording that is the source of the fragment [1]. Nowadays, audio identification is widely used in commercial systems such as Shazam.[3] Typically, the query fragment is exposed to signal distortions on the transmission channel [8, 29]. Recent identification algorithms exhibit a high degree of robustness against noise, MP3 compression artifacts, uniform temporal distortions, or interferences of multiple signals [16, 22]. The high specificity of this retrieval task goes along with a notion of similarity that is very close to the identity. To make this point clearer, we distinguish between a piece of music (in an abstract sense) and a specific performance of this piece. In particular for Western classical music, there typically exist a large number of different recordings of the same piece of music performed by different musicians. Given a query fragment, e. g., taken from a Bernstein recording of Beethoven's Symphony No. 5, audio fingerprinting systems are not capable of retrieving, e. g., a Karajan recording of the same piece. Likewise, given a query fragment from a live performance of "Act naturally" by The Beatles, the original studio recording of this song may not be found. The reason for this is that existing fingerprinting algorithms are not designed to deal with strong non-linear temporal distortions or with other musically motivated variations that affect, for example, the tempo or the instrumentation.

At a lower specificity level, the goal of fragment-based *audio matching* is to retrieve all audio fragments that musically correspond to a query fragment from all audio documents contained in a given database [28, 37]. In this scenario, one explicitly allows semantically motivated variations as they typically occur in different performances and arrangements of a piece of music. These variations include significant non-linear global and local differences in tempo, articulation, and phrasing as well as differences in executing note groups such as grace notes, trills, or arpeggios. Furthermore, one has to deal with considerable dynamical and spectral variations, which result from differences in instrumentation and loudness.

One instance of document-level retrieval at a similar specificity level as audio matching is the task of *version identification*. Here, the goal is to identify different versions of the same piece of music within a database [42]. In this scenario, one not only deals with changes in instrumentation, tempo, and tonality, but also with more extreme variations concerning the musical structure, key, or melody, as typically occurring in remixes and cover songs. This requires document-level similarity measures to globally compare entire documents.

Finally, there are a number of even less specific document-level retrieval tasks which

[3] `www.shazam.com` (accessed Dec. 18, 2011)

can be grouped under the term *category-based retrieval*. This term encompasses retrieval of documents whose relationship can be described by cultural or musicological categories. Typical categories are genre [50], rhythm styles [19, 41], or mood and emotions [26, 47, 53] and can be used in fragment as well as document-level retrieval tasks. Music recommendation or general music similarity assessments [7, 54] can be seen as further document-level retrieval tasks of low specificity.

In the following, we elaborate the aspects of specificity and granularity by means of representative state-of-the-art content-based retrieval approaches. In particular, we highlight characteristics and differences in requirements when designing and implementing systems for audio identification, audio matching, and version identification. Furthermore, we address efficiency and scalability issues. We start with discussing high-specific audio fingerprinting (Section 2), continue with mid-specific audio matching (Section 3), and then discuss version identification (Section 4). In Section 5, we discuss open problems in the field of content-based retrieval and give an outlook on future directions.

2 Audio Identification

Of all content-based music retrieval tasks, audio identification has received most interest and is now widely used in commercial applications. In the identification process, the audio material is compared by means of so-called *audio fingerprints*, which are compact content-based signatures of audio recordings [8]. In real-world applications, these fingerprints need to fulfill certain requirements. First of all, the fingerprints should capture highly specific characteristics so that a short audio fragment suffices to reliably identify the corresponding recording and distinguish it from millions of other songs. However, in real-world scenarios, audio signals are exposed to distortions on the transmission channel. In particular, the signal is likely to be affected by noise, artifacts from lossy audio compression, pitch shifting, time scaling, equalization, or dynamics compression. For a reliable identification, fingerprints have to show a significant degree of robustness against such distortions. Furthermore, scalability is an important issue for all content-based retrieval applications. A reliable audio identification system needs to capture the entire digital music catalog, which is further growing every day. In addition, to minimize storage requirements and transmission delays, fingerprints should be compact and efficiently computable [8]. Most importantly, this also requires efficient retrieval strategies to facilitate very fast database look-ups. These requirements are crucial for the design of large-scale audio identification systems. To satisfy all these requirements, however, one typically has to face a trade-off between contradicting principles.

There are various ways to design and compute fingerprints. One group of fingerprints consist of short sequences of frame-based feature vectors such as Mel-Frequency Cepstral Coefficients (MFCC) [9], Bark-scale spectrograms [22, 23], or a set of low-level descriptors [1]. For such representations, vector quantization [1] or thresholding [22] techniques, or temporal statistics [38] are needed for obtaining the required robustness. Another group of fingerprints consist of a sparse set of characteristic points such as spectral peaks [14, 52] or characteristic wavelet coefficients [24]. As an example, we now describe the peak-based fingerprints suggested by Wang [52], which are now commercially used in the Shazam music identification service[4].

The Shazam system provides a smartphone application that allows users to record a short audio fragment of an unknown song using the built-in microphone. The application

[4] `www.shazam.com` (accessed Dec. 18, 2011)

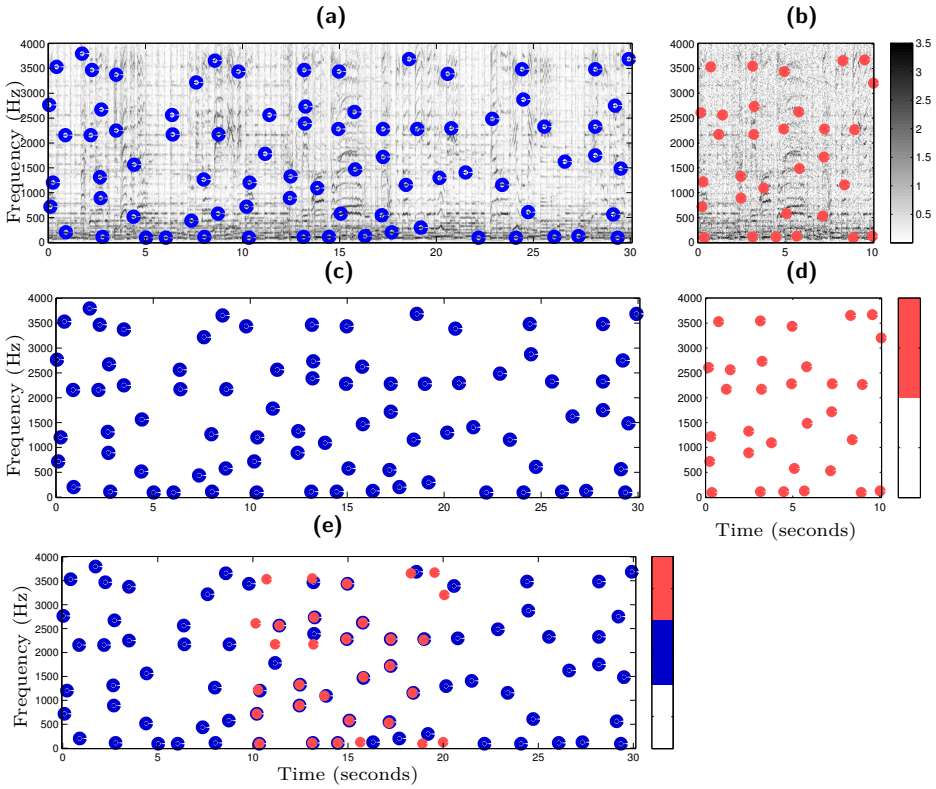

Figure 3 Illustration of the Shazam audio identification system using a recording of "Act naturally" by The Beatles as example. **(a)** Database document with extracted peak fingerprints. **(b)** Query fragment (10 seconds) with extracted peak fingerprints. **(c)** Constellation map of database document. **(d)** Constellation map of query document. **(e)** Superposition of the database fingerprints and time-shifted query fingerprints.

then derives the audio fingerprints which are sent to a server that performs the database look-up. The retrieval result is returned to the application and presented to the user together with additional information about the identified song. In this approach, one first computes a spectrogram from an audio recording using a short-time Fourier transform. Then, one applies a peak-picking strategy that extracts local maxima in the magnitude spectrogram: time-frequency points that are locally predominant. Figure 3 illustrates the basic retrieval concept of the Shazam system using a recording of "Act naturally" by The Beatles. Figure 3a and Figure 3b show the spectrogram for an example database document (30 seconds of the recording) and a query fragment (10 seconds), respectively. The extracted peaks are superimposed to the spectrograms. The peak-picking step reduces the complex spectrogram to a "constellation map", a low-dimensional sparse representation of the original signal by means of a small set of time-frequency points, see Figure 3c and Figure 3d. According to [52], the peaks are highly characteristic, reproducible, and robust against many, even significant distortions of the signal. Note that a peak is only defined by its time and frequency values, whereas magnitude values are no longer considered.

The general database look-up strategy works as follows. Given the constellation maps for a query fragment and all database documents, one locally compares the query fragment to all database fragments of the same size. More precisely, one counts matching peaks, i. e., peaks that occur in both constellation maps. A high count indicates that the corresponding database fragment is likely to be a correct hit. This procedure is illustrated in Figure 3e,

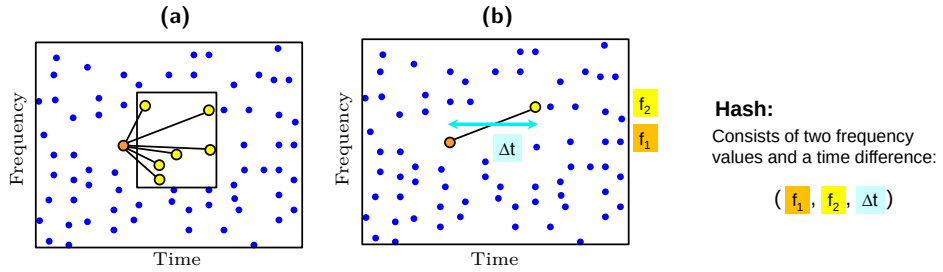

Figure 4 Illustration of the peak pairing strategy of the Shazam algorithm. **(a)** Anchor peak and assigned target zone. **(b)** Pairing of anchor peak and target peaks to form hash values.

showing the superposition of the database fingerprints and time-shifted query fingerprints. Both constellation maps show a high consistency (many red and blue points coincide) at a fragment of the database document starting at time position 10 seconds, which indicates a hit. However, note that not all query and database peaks coincide. This is because the query was exposed to signal distortions on the transmission channel (in this example additive white noise). Even under severe distortions of the query, there still is a high number of coinciding peaks thus showing the robustness of these fingerprints.

Obviously, such an exhaustive search strategy is not feasible for a large database as the run-time linearly depends on the number and sizes of the documents. For the constellation maps, as proposed in [29], one tries to efficiently reduce the retrieval time using indexing techniques—very fast operations with a sub-linear run-time. However, directly using the peaks as hash values is not possible as the temporal component is not translation-invariant and the frequency component alone does not have the required specificity. In [52], a strategy is proposed, where one considers pairs of peaks. Here, one first fixes a peak to serve as "anchor peak" and then assigns a "target zone" as indicated in Figure 4a. Then, pairs are formed of the anchor and each peak in the target zone, and a hash value is obtained for each pair of peaks as a combination of both frequency values and the time difference between the peaks as indicated in Figure 4b. Using every peak as anchor peak, the number of items to be indexed increases by a factor that depends on the number of peaks in the target zone. This combinatorial hashing strategy has three advantages. Firstly, the resulting fingerprints show a higher specificity than single peaks, leading to an acceleration of the retrieval as fewer exact hits are found. Secondly, the fingerprints are translation-invariant as no absolute timing information is captured. Thirdly, the combinatorial multiplication of the number of fingerprints introduced by considering pairs of peaks as well as the local nature of the peak pairs increases the robustness to signal degradations.

The Shazam audio identification system facilitates a high identification rate, while scaling to large databases. One weakness of this algorithm is that it can not handle time scale modifications of the audio as frequently occurring in the context of broadcasting monitoring. The reason for this is that time scale modifications (also leading to frequency shifts) of the query fragment completely change the hash values. Extensions of the original algorithms dealing with this issue are presented in [14, 51].

3 Audio Matching

The problem of audio identification can be regarded as largely solved even for large scale music collections. Less specific retrieval tasks, however, are still mostly unsolved. In this

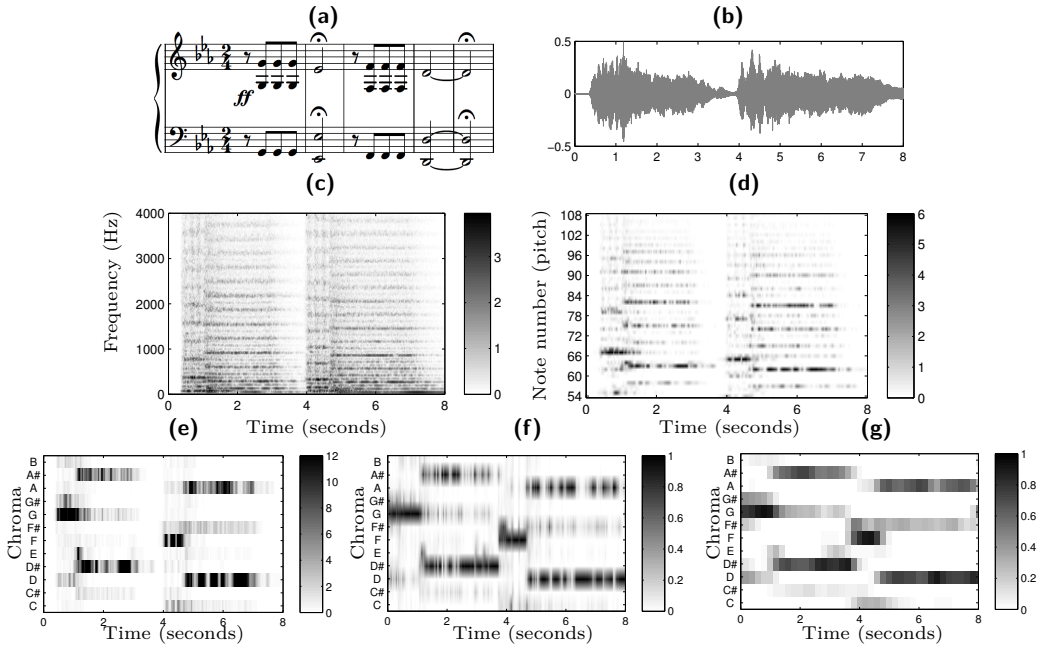

Figure 5 Illustration of various feature representations for the beginning of Beethoven's Opus 67 (Symphony No. 5) in a Bernstein interpretation. **(a)** Score of the excerpt. **(b)** Waveform. **(c)** Spectrogram with linear frequency axis. **(d)** Spectrogram with frequency axis corresponding to musical pitches. **(e)** Chroma features. **(f)** Normalized chroma features. **(g)** Smoothed version of chroma features, see also [36].

section, we highlight the difference between high-specific audio identification and mid-specific audio matching while presenting strategies to cope with musically motivated variations. In particular, we introduce chroma-based audio features [2, 17, 34] and sketch distance measures that can deal with local tempo distortions. Finally, we indicate how the matching procedure may be extended using indexing methods to scale to large datasets [10, 28].

For the audio matching task, suitable descriptors are required to capture characteristics of the underlying piece of music, while being invariant to properties of a particular recording. Chroma-based audio features [2, 34], sometimes also referred to as pitch class profiles [17], are a well-established tool for analyzing Western tonal music and have turned out to be a suitable mid-level representation in the retrieval context [10, 28, 37, 34]. Assuming the equal-tempered scale, the chroma attributes correspond to the set $\{\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots, \mathrm{B}\}$ that consists of the twelve pitch spelling attributes as used in Western music notation. Capturing energy distributions in the twelve pitch classes, chroma-based audio features closely correlate to the harmonic progression of the underlying piece of music. This is the reason why basically every matching procedure relies on some type of chroma feature.

There are many ways for computing chroma features. For example, the decomposition of an audio signal into a chroma representation (or chromagram) may be performed either by using short-time Fourier transforms in combination with binning strategies [17] or by employing suitable multirate filter banks [34, 36]. Figure 5 illustrates the computation of chroma features for a recording of the first five measures of Beethoven's Symphony No. 5 in a Bernstein interpretation. The main idea is that the fine-grained (and highly specific) signal representation as given by a spectrogram (Figure 5c) is coarsened in a musically meaningful way. Here, one adapts the frequency axis to represent the semitones of the equal tempered scale (Figure 5d). The resulting representation captures musically relevant pitch

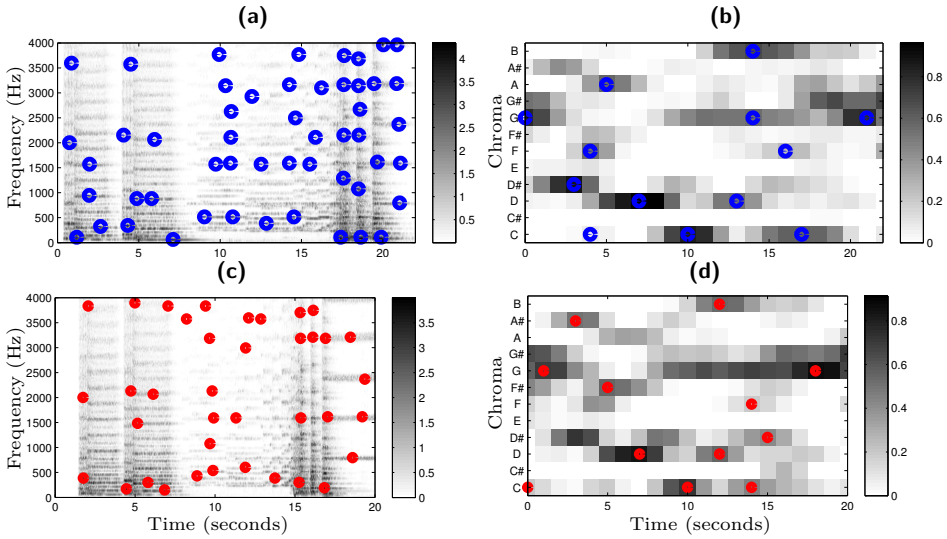

Figure 6 Different representations and peak fingerprints extracted for recordings of the first 21 measures of Beethoven's Symphony No. 5. **(a)** Spectrogram-based peaks for a Bernstein recording. **(b)** Chromagram-based peaks for a Bernstein recording. **(c)** Spectrogram-based peaks for a Karajan recording. **(d)** Chromagram-based peaks for a Karajan recording.

information of the underlying music piece, while being significantly more robust against spectral distortions than the original spectrogram. To obtain chroma features, pitches differing by octaves are summed up to yield a single value for each pitch class, see Figure 5e. The resulting chroma features show increased robustness against changes in timbre, as typically resulting from different instrumentations.

The degree of robustness of the chroma features against musically motivated variations can be further increased by using suitable post-processing steps. See [36] for some chroma variants.[5] For example, normalizing the chroma vectors (Figure 5f) makes the features invariant to changes in loudness or dynamics. Furthermore, applying a temporal smoothing and downsampling step (see Figure 5g) may significantly increase robustness against local temporal variations that typically occur as a result of local tempo changes or differences in phrasing and articulation. There are many more variants of chroma features comprising various processing steps. For example, applying logarithmic compression or whitening procedures enhances small yet perceptually relevant spectral components and the robustness to timbre [33, 35]. A peak picking of spectrum's local maxima can enhance harmonics while suppressing noise-like components [17, 13]. Furthermore, generalized chroma representations with 24 or 36 bins (instead of the usual 12 bins) allow for dealing with differences in tuning [17]. Such variations in the feature extraction pipeline have a large influence and the resulting chroma features can behave quite differently in the subsequent analysis task.

Figure 6 shows spectrograms and chroma features for two different interpretations (by Bernstein and Karajan) of Beethoven's Symphony No. 5. Obviously, the chroma features exhibit a much higher similarity than the spectrograms, revealing the increased robustness against musical variations. The fine-grained spectrograms, however, reveal characteristics of the individual interpretations. To further illustrate this, Figure 6 also shows fingerprint peaks

[5] MATLAB implementations for some chroma variants are supplied by the Chroma Toolbox: `www.mpi-inf.mpg.de/resources/MIR/chromatoolbox` (accessed Dec. 18, 2011)

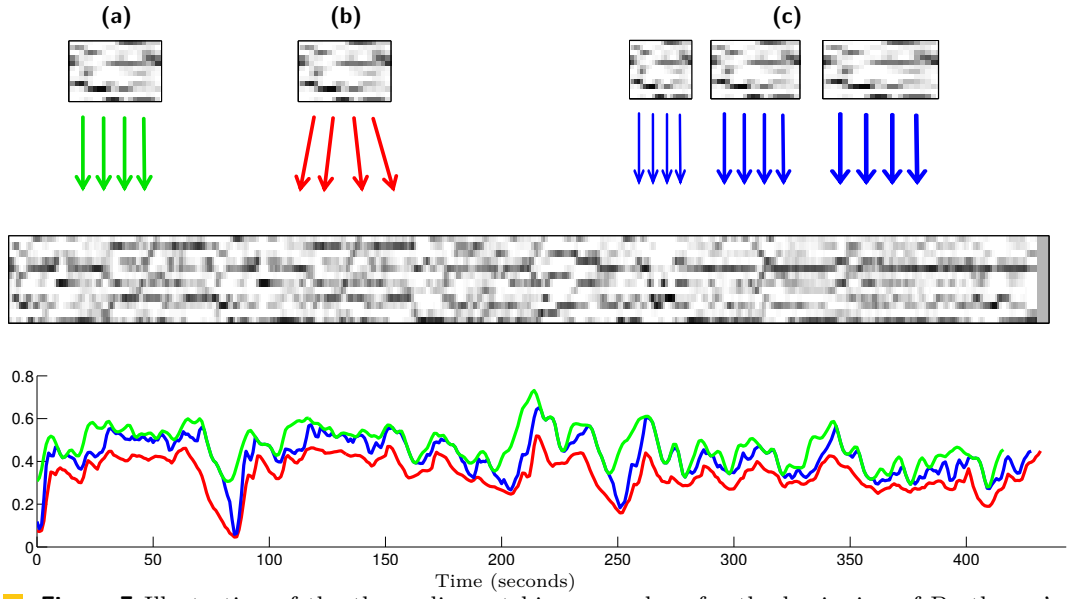

Figure 7 Illustration of the the audio matching procedure for the beginning of Beethoven's Opus 67 (Symphony No. 5) using a query fragment corresponding to the first 22 seconds (measures 1-21) of a Bernstein interpretation and a database consisting of an entire recording of a Karajan interpretation. Three different strategies are shown leading to three different matching curves. **(a)** Strict subsequence matching. **(b)** DTW-based matching. **(c)** Multiple query scaling strategy.

for all representations. As expected, the spectrogram peaks are very inconsistent for the different interpretations. The chromagram peaks, however, show at least some consistencies, indicating that fingerprinting techniques could also be applicable for audio matching [6]. In practice, however, the fragile peak picking step on the basis of the rather coarse chroma features may not lead to robust results. Furthermore, one has to find a technique to deal with the local and global tempo differences between the interpretations. See [21] for a detailed investigation of this approach.

Instead of using sparse peak representations, one typically employs a subsequence search, which is directly performed on the chroma features. Here, a query chromagram is compared with all subsequences of database chromagrams. As a result one obtains a matching curve as shown in Figure 7, where a small value indicates that the subsequence of the database starting at this position is similar to the query sequence. Then the best match is the minimum of the matching curve. In this context, one typically applies distance measures that can deal with tempo differences between the versions, such as edit distances [3], dynamic time warping (DTW) [34, 37], or the Smith-Waterman algorithm [43]. An alternative approach is to linearly scale the query to simulate different tempi and then to minimize over the distances obtained for all scaled variants [28]. Figure 7 shows three different matching curves which are obtained using strict subsequence matching, DTW, and a multiple query strategy.

To speed up such exhaustive matching procedures, one requires methods that allow for efficiently detecting *near* neighbors rather than exact matches. A first approach in this direction uses inverted file indexing [28] and depends on a suitable codebook consisting of a finite set of characteristic chroma vectors. Such a codebook can be obtained in an unsupervised way using vector quantization or in a supervised way exploiting musical knowledge about chords. The codebook then allows for classifying the chroma vectors of the database and to index the vectors according to the assigned codebook vector. This results in

an inverted list for each codebook vector. Then, an exact search can be performed efficiently by intersecting suitable inverted lists. However, the performance of the exact search using quantized chroma vectors greatly depends on the codebook. This requires fault-tolerance mechanisms which partly eliminate the speed-up obtained by this method. Consequently, this approach is only applicable for databases of medium size [28]. An approach presented in [10] uses an index-based near neighbor strategy based on locality sensitive hashing (LSH). Instead of considering long feature sequences, the audio material is split up into small overlapping *shingles* that consist of short chroma feature subsequences. The shingles are then indexed using locality sensitive hashing which allows for scaling this approach to larger datasets. However, to cope with temporal variations, each shingle covers only a small portion of the audio material and queries need to consist of a large number of shingles. The high number of table look-ups induced by this strategy may become problematic for very large datasets where the index is stored on a secondary storage device. The approach presented in [20] is also based on LSH. However, to reduce the number of table look-ups, each query consists of only a single shingle covering 15–25 seconds of the audio. To handle temporal variations, a combination of local feature smoothing and global query scaling is proposed.

In summary, mid-specific audio matching using a combination of highly robust chroma features and sequence-based similarity measures that account for different tempi results in a good retrieval quality. However, the low specificity of this task makes indexing much harder than in the case of audio identification. This task becomes even more challenging when dealing with relatively short fragments on the query and database side.

4 Version Identification

In the previous tasks, a musical fragment is used as query and similar fragments or documents are retrieved according to a given degree of specificity. The degree of specificity was very high for audio identification and more relaxed for audio matching. If we allow for even less specificity, we are facing the problem of version identification [42]. In this scenario, a user wants to retrieve not only exact or near-duplicates of a given query, but also any existing re-interpretation of it, no matter how radical such a re-interpretation might be. In general, a version may differ from the original recording in many ways, possibly including significant changes in timbre, instrumentation, tempo, main tonality, harmony, melody, and lyrics. For example, in addition to the aforementioned Karajan's rendition of Beethoven's Symphony No. 5, one could be also interested in a live performance of it, played by a punk-metal band who changes the tempo in a non-uniform way, transposes the piece to another key, and skips many notes as well as most parts of the original structure. These types of documents where, despite numerous and important variations, one can still unequivocally glimpse the original composition are the ones that motivate version identification.

Version identification is usually interpreted as a document-level retrieval task, where a single similarity measure is considered to globally compare entire documents [3, 13, 46]. However, successful methods perform this global comparison on a local basis. Here, the final similarity measure is inferred from locally comparing only parts of the documents—a strategy that allows for dealing with non-trivial structural changes. This way, comparisons are performed either on some representative part of the piece [18], on short, randomly chosen subsequences of it [32], or on the best possible longest matching subsequence [43, 44].

A common approach to version identification starts from the previously introduced chroma features; also more general representations of the tonal content such as chords or tonal templates have been used [42]. Furthermore, melody-based approaches have been

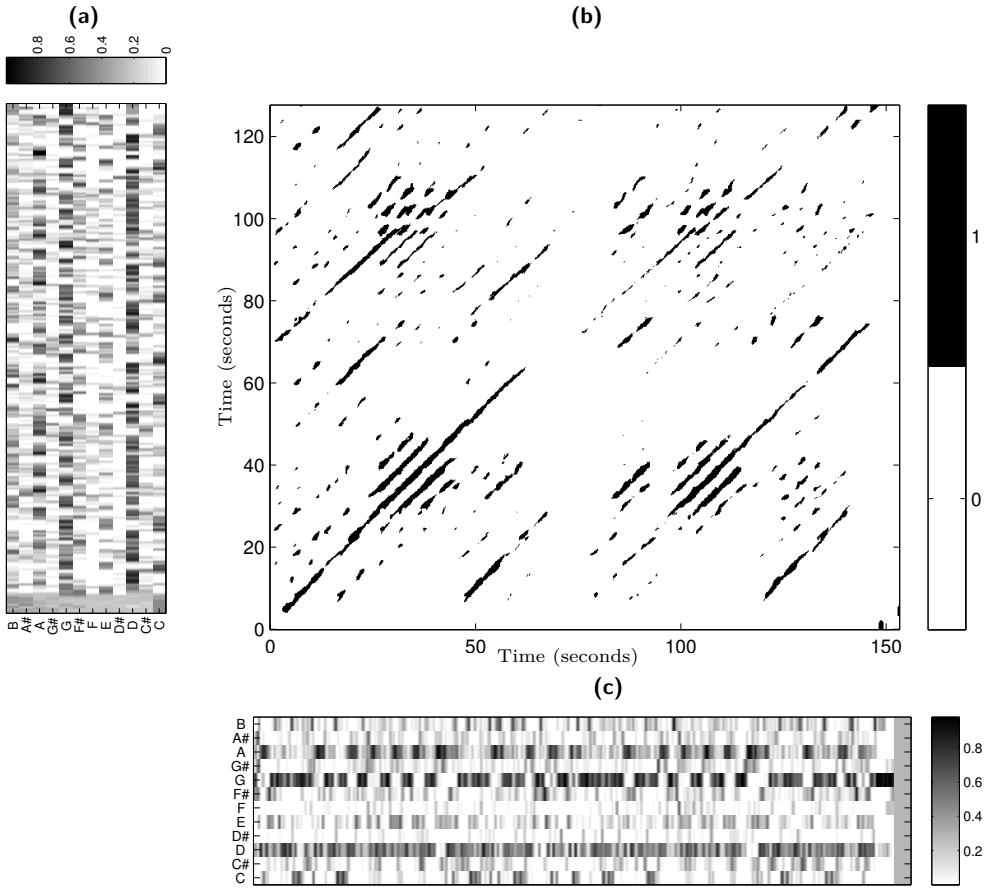

Figure 8 Similarity matrix for "Act naturally" by The Beatles, which is actually a cover version of a song by Buck Owens. **(a)** Chroma features of the version by The Beatles. **(b)** Score matrix. **(c)** Chroma features of the version by Buck Owens.

suggested, although recent findings suggest that this representation may be suboptimal [15, 40]. Once a tonal representation is extracted from the audio, changes in the main tonality need to be tackled, either in the extraction phase itself, or when performing pairwise comparisons of such representations.

Tempo and timing deviations have a strong effect in the chroma feature sequences, hence making their direct pairwise comparison problematic. An intuitive way to deal with global tempo variations is to use beat-synchronous chroma representations [6, 13]. However, the required beat tracking step is often error-prone for certain types of music and therefore may negatively affect the final retrieval result. Again, as for the audio matching task, dynamic programming algorithms are a standard choice for dealing with tempo variations [34], this time applied in a local fashion to identify longest matching subsequences or local alignments [43, 44].

An example of such an alignment procedure is depicted in Figure 8 for our "Act naturally" example by The Beatles. The chroma features of this version are shown in Figure 8c. Actually, this song is originally not written by The Beatles but a cover version of a Buck Owens song of the same name. The chroma features of the original version are shown in Figure 8a. Alignment algorithms rely on some sort of scores (and penalties) for matching (mismatching) individual chroma sequence elements. Such scores can be real-valued or binary. Figure 8b shows a binary score matrix encoding pair-wise similarities between chroma vectors of the two sequences. The binarization of score values provides some additional robustness against

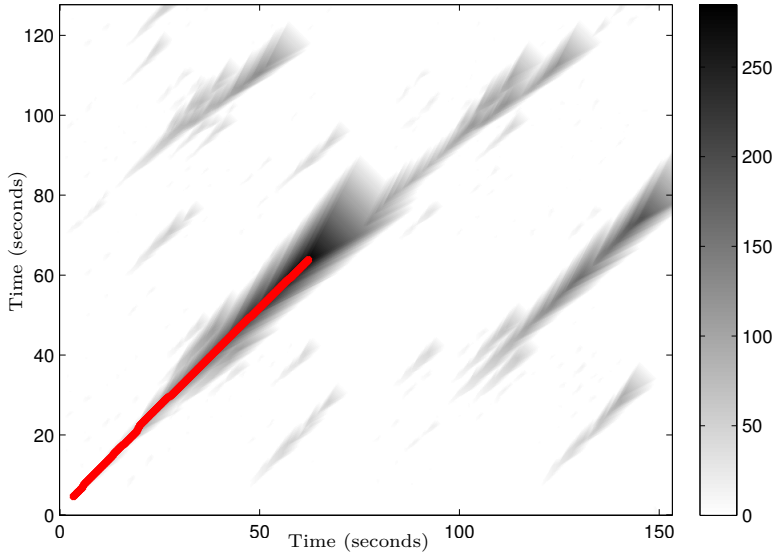

Figure 9 Accumulated score matrix with optimal alignment path for the "Act naturally" example (as shown in Figure 8).

small spectral and tonal differences. Correspondences between versions are revealed by the score matrix in the form of diagonal paths of high score. For example, in Figure 8, one observes a diagonal path indicating that the first 60 seconds of the two versions exhibit a high similarity.

For detecting such path structures, dynamic programming strategies make use of an accumulated score matrix. In their local alignment version, where one is searching for subsequence correspondences, this matrix reflects the lengths and quality of such matching subsequences. Each element (consisting of a pair of indices) of the accumulated score matrix corresponds to the end of a subsequence and its value encodes the score accumulated over all elements of the subsequence. Figure 9 shows an example of the accumulated score matrix obtained for the score matrix in Figure 8. The highest-valued element of the accumulated score matrix corresponds to the end of the most similar matching subsequence. Typically, this value is chosen as the final score for the document-level comparison of the two pieces. Furthermore, the specific alignment path can be easily obtained by backtracking from this highest element [34]. The alignment path is indicated by the red line in Figure 9. Additional penalties account for the importance of insertions/deletions in the subsequences. In fact, the way of deriving these scores and penalties is usually an important part of the version identification algorithms and different variants have been proposed [3, 43, 44]. The aforementioned final score is directly used for ranking candidate documents to a given query. It has recently been shown that such rankings can be improved by combining different scores obtained by different methods [39], and by exploiting the fact that alternative renditions of the same piece naturally cluster together [30, 45].

The task of version identification allows for these and many other new avenues for research [42]. However, one of the most challenging problems that remains to be solved is to achieve high accuracy and scalability at the same time, allowing low-specific retrieval in large music collections [6]. Unfortunately, the accuracies achieved with today's non-scalable approaches have not yet been reached by the scalable ones, the latter remaining far behind the former.

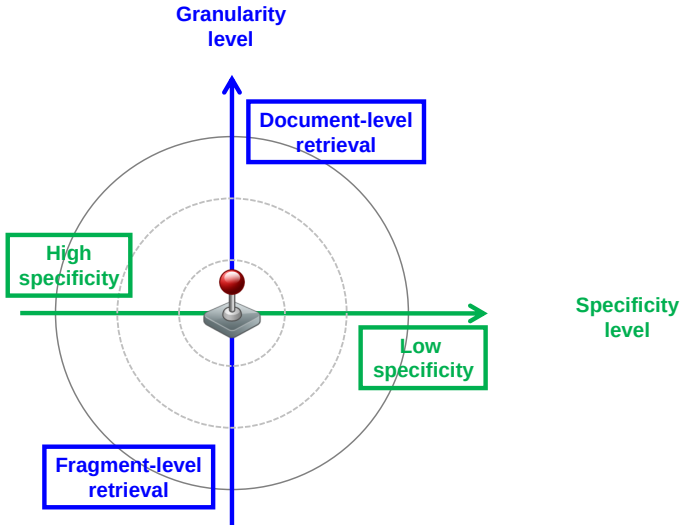

Figure 10 Joystick-like user interface for continuously adjusting the specificity and granularity levels used in the retrieval process.

5 Outlook

In this paper, we have discussed three representative retrieval strategies based on the query-by-example paradigm. Such content-based approaches provide mechanisms for discovering and accessing music even in cases where the user does not explicitly know what he or she is actually looking for. Furthermore, such approaches complement traditional approaches that are based on metadata and tags. The considered level of specificity has a significant impact on the implementation and efficiency of the retrieval system. In particular, search tasks of high specificity typically lead to exact matching problems, which can be realized efficiently using indexing techniques. In contrast, search tasks of low specificity need more flexible and cost-intensive mechanisms for dealing with spectral, temporal, and structural variations. As a consequence, the scalability to huge music collections comprising millions of songs still poses many yet unsolved problems.

Besides efficiency issues, one also has to better account for user requirements in content-based retrieval systems. For example, one may think of a comprehensive framework that allows a user to adjust the specificity level at any stage of the search process. Here, the system should be able to seamlessly change the retrieval paradigm from high-specific audio identification, over mid-specific audio matching and version identification to low-specific genre identification. Similarly, the user should be able to flexibly adapt the granularity level to be considered in the search. Furthermore, the retrieval framework should comprise control mechanisms for adjusting the musical properties of the employed similarity measure to facilitate searches according to rhythm, melody, or harmony or any combination of these aspects.

Figure 10 illustrates a possible user interface for such an integrated content-based retrieval framework, where a joystick allows a user to continuously and instantly adjust the retrieval specificity and granularity. For example, a user may listen to a recording of Beethoven's Symphony No. 5, which is first identified to be a Bernstein recording using an audio identification strategy (moving the joystick to the leftmost position). Then, being interested in different

versions of this piece, the user moves the joystick upwards (document-level) and to the right (mid-specific), which triggers a version identification. Subsequently, shifting towards a more detailed analysis of the piece, the user selects the famous fate motif as query and moves the joystick downwards to perform some mid-specific fragment-based audio matching. Then, the system returns the positions of all occurrences of the motif in all available interpretations. Finally, moving the joystick to the rightmost position, the user may discover recordings of pieces that exhibit some general similarity like style or mood. In combination with immediate visualization, navigation, and feedback mechanisms, the user is able to successively refine and adjust the query formulation as well as the retrieval strategy, thus leading to novel strategies for exploring, browsing, and interacting with large collections of audio content.

Another major challenge refers to cross-modal music retrieval scenarios, where the query as well as the retrieved documents can be of different modalities. For example, one might use a small fragment of a musical score to query an audio database for recordings that are related to this fragment. Or a short audio fragment might be used to query a database containing MIDI files. In the future, comprehensive retrieval frameworks are to be developed that offer multi-faceted search functionalities in heterogeneous and distributed music collections containing all sorts of music-related documents.

6 Acknowledgment

We would like to express our gratitude to Christian Dittmar, Emilia Gómez, Frank Kurth, and Markus Schedl for their helpful and constructive feedback.

References

1 Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, and Markus Cremer. AudioID: Towards content-based identification of audio material. In *Proceedings of the 110th AES Convention*, Amsterdam, NL, 2001.

2 Mark A. Bartsch and Gregory H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, February 2005.

3 Juan Pablo Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 239–244, Vienna, Austria, 2007.

4 Thierry Bertin-Mahieux, Douglas Eck, Francois Maillet, and Paul Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

5 Thierry Bertin-Mahieux, Douglas Eck, and Michael I. Mandel. Automatic tagging of audio: The state-of-the-art. In Wenwu Wang, editor, *Machine Audition: Principles, Algorithms and Systems*, chapter 14, pages 334–352. IGI Publishing, 2010.

6 Thierry Bertin-Mahieux and Daniel P.W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Platz, NY, 2011.

7 Dmitry Bogdanov, Joan Serrà, Nicolas Wack, and Perfecto Herrera. Unifying low-level and high-level music similarity measures. *IEEE Transactions on Multimedia*, 13(4):687 –701, aug. 2011.

8 Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, 41(3):271–284, 2005.

9 Pedro Cano, Eloi Batlle, Harald Mayer, and Helmut Neuschmied. Robust sound modeling for song detection in broadcast audio. In *Proceedings of the 112th AES Convention*, pages 1–7, 2002.

10 Michael A. Casey, Christophe Rhodes, and Malcolm Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech & Language Processing*, 16(5):1015–1028, 2008.

11 Michael A. Casey, Remco Veltkap, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

12 Òscar Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 1st edition, September 2010.

13 Daniel P.W. Ellis and Graham E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 1429–1432, Honolulu, Hawaii, USA, April 2007.

14 Sébastien Fenet, Gaël Richard, and Yves Grenier. A scalable audio fingerprint method with robustness to pitch-shifting. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, USA, 2011.

15 Rémi Foucard, Jean-Louis Durrieu, Mathieu Lagrange, and Gaël Richard. Multimodal similarity between musical streams for cover version detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5514–5517, Dallas, USA, 2010.

16 Dimitrios Fragoulis, George Rousopoulos, Thanasis Panagopoulos, Constantin Alexiou, and Constantin Papaodysseus. On the automated recognition of seriously distorted musical recordings. *IEEE Transactions on Signal Processing*, 49(4):898–908, 2001.

17 Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, UPF Barcelona, 2006.

18 Emilia Gómez, Bee Suan Ong, and Perfecto Herrera. Automatic tonal analysis from music summaries for version identification. In *Proceedings of the 121st AES Convention*, San Francisco, CA, USA, 2006.

19 Fabien Gouyon. *A computational approach to rhythm description: audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2005.

20 Peter Grosche and Meinard Müller. Toward characteristic audio shingles for efficient cross-version music retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 473–476, Kyoto, Japan, 2012.

21 Peter Grosche and Meinard Müller. Toward musically-motivated audio fingerprints. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 93–96, Kyoto, Japan, 2012.

22 Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 107–115, Paris, France, 2002.

23 Jaap Haitsma and Ton Kalker. Speed-change resistant audio fingerprinting using auto-correlation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 728–731, 2003.

24 Yan Ke, Derek Hoiem, and Rahul Sukthankar. Computer vision for music identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 597–604, San Diego, CA, USA, 2005.

25 Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. John Wiley & Sons, 2005.

26 Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon C. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 255–266, Utrecht, The Netherlands, 2010.

27 Aniket Kittur, Ed Chi, Bryan A. Pendleton, Bongwon Suh, and Todd Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. In *Computer/Human Interaction Conference (Alt.CHI)*, San Jose, CA, 2007.

28 Frank Kurth and Meinard Müller. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, February 2008.

29 Frank Kurth, Andreas Ribbrock, and Michael Clausen. Identification of highly distorted audio material for querying large scale data bases. In *Proceedings of the 112th AES Convention*, 2002.

30 Mathieu Lagrange and Joan Serrà. Unsupervised accuracy improvement for cover song detection using spectral connectivity network. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 595–600, 2010.

31 Paul Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008.

32 Matija Marolt. A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia*, 10(8):1617–1625, dec. 2008.

33 Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–140, 2010.

34 Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

35 Meinard Müller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.

36 Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 215–220, Miami, USA, 2011.

37 Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 288–295, 2005.

38 Mathieu Ramona and Geoffroy Peeters. Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 477–480, 2011.

39 Suman Ravuri and Daniel P.W. Ellis. Cover song detection: from high scores to general classification. In *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, pages 65–68, Dallas, TX, 2010.

40 Justin Salamon, Joan Serrà, and Emilia Gómez. Melody, bass line and harmony descriptions for music version identification. In *preparation*, 2011.

41 Björn Schuller, Florian Eyben, and Gerhard Rigoll. Tango or waltz?: Putting ballroom dance style into tempo detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2008:12, 2008.

42 Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation and beyond. In Z. W. Ras and A. A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, volume 16 of *Studies in Computational Intelligence*, chapter 14, pages 307–332. Springer, Berlin, Germany, 2010.

43 Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1138–1151, oct 2008.

44 Joan Serrà, Xavier Serra, and Ralph G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.

45 Joan Serrà, Massimiliano Zanin, Perfecto Herrera, and Xavier Serra. Characterization and exploitation of community structure in cover song networks. *Pattern Recognition Letters*, 2010. Submitted.

46 Wei-Ho Tsai, Hung-Ming Yu, and Hsin-Min Wang. Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *Journal of Information Science and Engineering*, 24(6):1669–1687, 2008.

47 Emiru Tsunoo, Taichi Akase, Nobutaka Ono, and Shigeki Sagayama. Musical mood classification by rhythm and bass-line unit pattern analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2010.

48 Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Five approaches to collecting tags for music. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 225–230, Philadelphia, USA, 2008.

49 Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):467–476, 2008.

50 George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

51 Jan Van Balen. Automatic recognition of samples in musical audio. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2011.

52 Avery Wang. An industrial strength audio search algorithm. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 7–13, Baltimore, USA, 2003.

53 Felix Weninger, Martin Wöllmer, and Björn Schuller. Automatic assessment of singer traits in popular music: Gender, age, height and race. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 37–42, Miami, Florida, USA, 2011.

54 Kris West and Paul Lamere. A model-based approach to constructing music similarity functions. *EURASIP Journal on Advances in Signal Processing*, 2007(1):024602, 2007.