Data-Driven Sound Track Generation*

Meinard Müller and Jonathan Driedger

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

Abstract

Background music is often used to generate a specific atmosphere or to draw our attention to specific events. For example in movies or computer games it is often the accompanying music that conveys the emotional state of a scene and plays an important role for immersing the viewer or player into the virtual environment. In view of home-made videos, slide shows, and other consumer-generated visual media streams, there is a need for computer-assisted tools that allow users to generate aesthetically appealing music tracks in an easy and intuitive way. In this contribution, we consider a data-driven scenario where the musical raw material is given in form of a database containing a variety of audio recordings. Then, for a given visual media stream, the task consists in identifying, manipulating, overlaying, concatenating, and blending suitable music clips to generate a music stream that satisfies certain constraints imposed by the visual data stream and by user specifications. It is our main goal to give an overview of various content-based music processing and retrieval techniques that become important in data-driven sound track generation. In particular, we sketch a general pipeline that highlights how the various techniques act together and come into play when generating musically plausible transitions between subsequent music clips.

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

Keywords and phrases Sound track, content-based retrieval, audio matching, time-scale modification, warping, tempo, beat tracking, harmony

Digital Object Identifier 10.4230/DFU.Vol3.11041.175

1 Introduction

The computer-assisted generation of sound tracks for given visual media streams has significantly gained in importance. For example, video games of these days are often accompanied by music of high artistic value and excellent sound quality coming close to sound tracks of movies. However, opposed to film music, the sound track underlying a video game has to constantly adapt to the respective scene of the game and to interactively react to the player's input.

When developing a high-quality computer game, composers are asked to create specific music clips that not only match the various scenes and characters of the game, but also account for transitions within and across different scenes. To this end, the music needs to contain various transition points that allow for smoothly connecting and bridging different passages at specified or even arbitrary points in time. Even though there may be no real

* This work has been supported by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI) and the German Research Foundation (DFG MU 2682/5-1). Meinard Müller is now with Bonn University, Department of Computer Science III, Germany.

Multimodal Music Processing. *Dagstuhl Follow-Ups, Vol. 3.* ISBN 978-3-939897-37-8.
Editors: Meinard Müller, Masataka Goto, and Markus Schedl; pp. 175–194

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

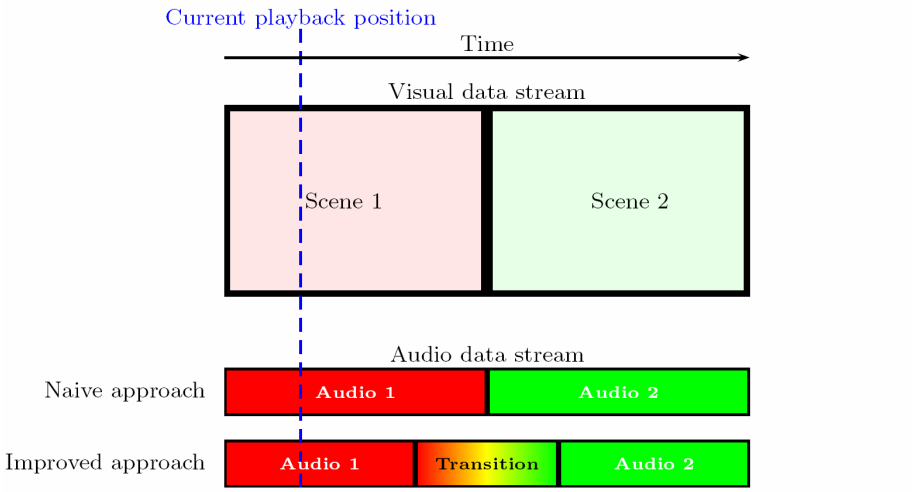

Figure 1 Sound track generation by concatenating existing audio clips.

alternatives to manually creating music in particular when artistic aspects are given priority, such *compositional approaches* to sound track generation are costly and labor extensive. Furthermore, the resulting music is highly specialized, and the system has a slow response time when transitions are possible only at specific pre-defined positions.

As an inexpensive alternative, one may revert to *parametric approaches*, where the background music is synthesized based on parametric models. Here, the free parameters allow for specifying, adjusting, and triggering sound events and may directly be controlled by scene annotations, by the moving objects within the scene, or by a user's input. However, even though offering fast response times, such parametric approaches may be aesthetically questionable from a musical point of view.

The main focus of this work follows a third approach. To obtain appealing sound tracks, one strategy is to simply play back an existing music recording that is in line with a given visual data stream. Reverting to an audio database comprising high-quality music recordings, the idea of such a *data-driven approach* is to identify and play back music clips that correspond well to the visual scenes while accounting for user specifications. However, a simple concatenation of audio clips may result in unpleasant and abrupt transitions between subsequent audio clips. Therefore, one main challenge consists in the creation of musically smooth and euphonious transitions, which are as pleasant as possible to the listener's ear, see Figure 1.

In this contribution, our main goal is to describe a possible pipeline for such a data-driven sound track generation system while giving an overview of the necessary data processing and retrieval techniques, see Figure 2. In the following, we exemplarily consider an online scenario, where a visual data stream, which consists of a sequence of changing scenes that are associated to certain categories (e. g., moods), is given. Furthermore, a comprehensive music database that contains audio recordings of various genres, styles and moods serves as basis for the sound track to be generated. These recordings are assumed to be annotated with respect to the same categories as used to describe the visual scenes. For the current scene, a specific audio recording is played back. As soon as the next scene change is pending, the category of the subsequent scene as well as the tolerable delay for the transition needs to be known. The system then determines a suitable region in the current audio recording,

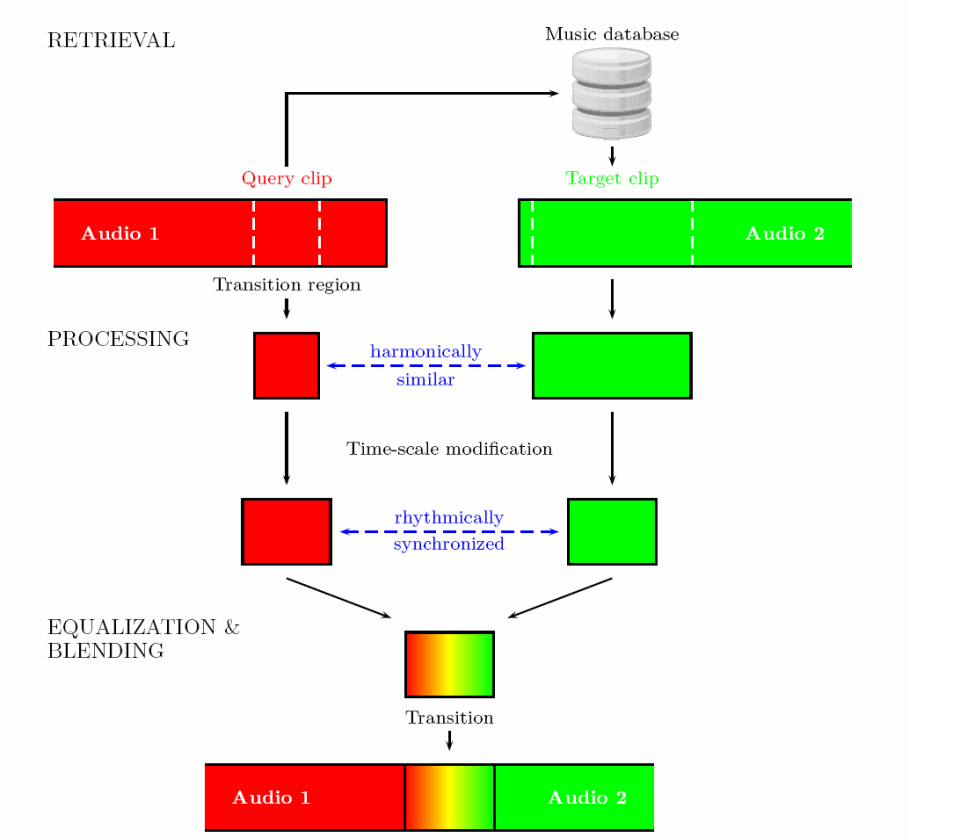

Figure 2 Overview of different retrieval and processing components required for a data-driven sound track generation system.

also referred to as *transition region*. The waveform corresponding to this region is then used as *query clip*, and content-based retrieval is performed to identify a suitable audio clip in the music database—referred to as *target clip*—satisfying the following two properties. Firstly, the clip should be contained in an audio recording that reflects the category of the subsequent scene. Secondly, the target clip should be similar to the query clip to allow for a smooth (e. g., harmonically and rhythmically plausible) transition. To this end, one particularly needs a clip that has a similar harmonic progression as the query clip. In the next step, the two clips are temporally synchronized by first estimating the beat positions and then applying suitable time-scale modifications (similar to what a DJ is doing). The actual transition from the current recording (containing the query clip) to the next recording (containing the target clip) is then realized by blending from the synchronized query clip to the target clip. Finally, to further improve the quality of the transition, one needs intelligent equalization techniques that can be used to attenuate possibly interfering sound components or to amplifying certain voices, instruments or notes.

In the sketched approach, various challenges arise. First, one needs similarity measures and content-based retrieval strategies to search for and identify suitable music clips that satisfy the given constraints. These constraints may not only be imposed by the visual input and user specifications, but also by algorithmic and aesthetic considerations. Furthermore, one requires a number of signal processing techniques that allow for adjusting the audio

material with respect to various musical aspects including harmony, rhythm, tempo, or polyphony. In the following, we give an overview of these techniques and provide suitable links to the literature. The remainder of this contribution is organized as follows. In Section 2, we discuss previous work that is related to the problem of automated sound track generation. Then, in Section 3, we give an overview of the involved data processing and retrieval techniques while highlighting how these techniques act together and come into play in a data-driven soundtrack generation scenario. Finally, we conclude with Section 4 discussing challenges, limitations, and future work.

2 Related Work

The idea of generating new music by concatenating existing music fragments based on euphonious transitions has a long history. At the end of the 18$^{\text{th}}$ century, "Musikalische Würfelspiele" ("Musical dice games") were a popular pastime, where a piano player had to create music by suitably concatenating measures from known pieces that were randomly chosen by throwing a dice [32, 22].

Nowadays the generation of dynamically changing music by concatenating pre-rendered music clips has become an important issue in particular in the context of video games. As emphasized in [70], the generation of suitable music can add emotional depth and soul to the various scenes leading to a highly immersive gaming experience. To this end, the music not only has to loosely reflect the mood of the respective scene, but also has to constantly adapt to, or even to anticipate the game's events and the player's actions. The term *adaptive audio* (or *adaptive music*) has been used to describe audio and music that responds appropriately to gameplay [70]. As one requirement, to support the game's continuity, music transitions that seamlessly connect the various moods and intensities are needed. To this end, one needs techniques that go far beyond a simple concatenation or cross-fade between subsequent audio clips. Instead, short building blocks of music, different layers (e. g., percussion loops, super-imposable melody and instrument tracks), as well as transitional cues are required for creating adaptive music. The composition of music that does not follow a linear flow (as is for traditional music) but that can be reassembled in a flexible and smooth fashion constitutes a hard problem—musically as well as technically. For a detailed discussion and further links, we also refer to [66].

There are various approaches to automatically generate music streams on the basis of *symbolic* music representations. For example, [11] describes an automated music generation system that works on the basis of MIDI files. Opposed to waveform-based audio representations, symbolic representations offer more flexibility and direct control since musical parameters such as note events, instrumentation, or tempo are given explicitly and can be therefore altered easily. On the downside, synthesizing music from a symbolic representation often leads to unsatisfying results, e. g., because of the artificiality of the used synthetic instruments or the lacking of performance nuances. Furthermore, high-quality symbolic representations are often not available or hard to generate from existing audio material.

The automated remixing and concatenation of existing audio material constitutes a challenging area of research. A prominent application scenario is what a disc jockey (DJ) is typically doing: he not only selects appropriate music for the audience, but also tries to mix and blend recorded music to create a continuous playback. First systems to automate this process are described, e. g., in [8, 37]. In the mixing process, DJs pay particular attention to a good rhythmical transition, which requires an adjustment of the tempo and a matching of the beats. A tool to automate the process of finding good rhythmical transitions is described

in [39]. Harmonic similarity of the two audio clips to be connected usually plays a minor role, even though professional DJs also often try to take the musical key into consideration. In [49], the authors describe a first system for concatenating audio clips to form a single long audio stream, where the recordings are ordered in such a way that euphonious transitions between the clips are possible. The positions of the transitions are chosen to maximize the local harmonic and rhythmic similarity of the two subsequent audio clips. This scenario is similar to what we want to consider in our contribution. However, we want to focus more on the underlying techniques that are need for realizing such a framework, whereas [49] describe a first overall system. Finally, we want to mention the work by [69], where the goal is to temporally rearrange a given music recording to fit certain user-specified constraints. In particular, suitable transitions points are identified within the audio material that allow for deleting, copying, and rearranging certain parts while keeping the flow of the music. This not only allows for an automated adjustment of the duration of a given recording but also for linking certain parts of the recording to specified key frames of the visual data stream.

3 Music Retrieval and Processing

We now give an overview of the various content-based music retrieval and processing techniques that are important in view of the described data-driven sound track generation scenario, see also Figure 12 for an overview. In particular, we have a focus on the creation of musically plausible transitions between audio recordings that are to be concatenated. To this end, one requires methods from audio signal analysis to capture harmonic and rhythmic properties of music recordings. Such properties form the basis for designing musically meaningful similarity measures needed to identify potential transition regions. Then, one requires manipulation techniques that allow for temporally (e. g., time-scaling, clipping) and spectrally (e. g., modulation, harmonic-percussive separation, voice equalization) manipulating the audio material. Furthermore, synthesis methods (e. g., blending, morphing) are needed to render the final audio stream. Last but not least, in view of efficiency and online capability, data structures are to be developed that facilitate fast content-based search and encode, for example, plausible transitions between music clips.

3.1 Category-based Classification

As mentioned in the introduction, we assume that the visual scenes are associated to certain categories that may refer to the emotional content or mood of the scene. For example, the current scene may be associated with the attribute "Angry" whereas the subsequent scene may by associated with the attribute "Happy." Then one important step in the sound track generation scenario is to find music recordings that reflect the categories of the given scenes, see Figure 3.

Actually, the automated classification of music recordings with respect to a given set of categories has been a central topic in the field of music information retrieval. Generally, such categories refer to cultural or musicological aspects [16] including genre [59, 63] or rhythm styles [26, 61]. In our scenario, we are particularly interested in categories that refer to mood or emotions [36, 41, 62]. However, as noted in [41], when organizing music in terms of emotional content, one is faced with the problem that there is a "considerable disagreement regarding the perception and interpretation of the emotions of a song or ambiguity within the piece itself." In other words, the categories are often ill-defined and highly subjective with the result that the automation of the classification problem is still in its early stages, see [41] for an overview.

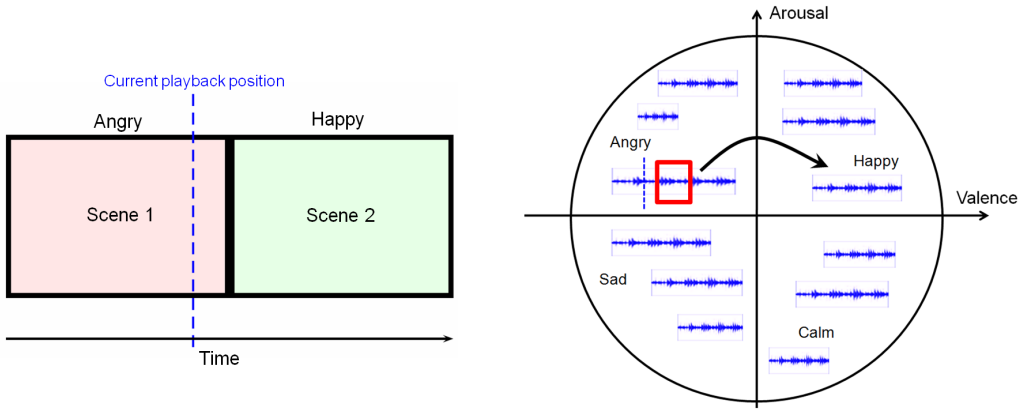

Figure 3 Visual scenes and music database annotated with respect to mood categories of the valence-arousal space [58].

In the following, we assume that the recordings of the database have been annotated according to given mood categories. Such annotations may be obtained by manual expert classification or may be derived from contextual text information (e. g., websites, tags, and lyrics) and content-based approaches [41]. In the music context, the most prominent way to organize emotional descriptors is the two-dimensional valence-arousal space as originally introduced in [58], see Figure 3. Here, the mood categories are arranged on a plane with two independent axes that encode arousal (intensity) ranging from low to high and valence (appraisal of polarity) ranging from negative to positive [41]. However, the specific nature of the descriptive labels and their organization is not in the scope of this contribution. In the following, we only require that both the visual scenes as well as the database documents are characterized based on the same set of categories.

3.2 Content-based Audio Retrieval

In our online scenario, we assume that a music recording is played back underlying the current visual scene. Once a scene change is pending and the category of the subsequent scene is known, the goal is to find a music recording which category fits the subsequent scene. Assuming suitable annotations as discussed in Section 3.1, this simply requires a table look-up to retrieve all music documents of the desired category. In addition, we want to generate a smooth transition from the current recording to the next one. Here, a simple cross-fade between two recordings may result in unpleasant listening experiences due to harmonic, melodic and rhythmic incompatibilities in the transition phase. Instead, the goal is to generate musically transitions that do not intercept the flow of the multimedia presentation. One way to achieve this goal is to specify a suitable region in the current audio recording, where the transition to the next recording is to be performed. Based on the corresponding clip, one then needs to identify a recording that contains a semantically related target clip allowing for a plausible transition. This is exactly the point, where content-based audio retrieval comes into play. In the following, we summarize two prominent retrieval scenarios and describe the techniques used in our pipeline.

Actually, in content-based audio retrieval, various levels of specificity can be considered. At the highest specificity level, the retrieval task is often referred to as *audio identification* or *audio fingerprinting*. Here, given a small audio fragment as query, the task consists in

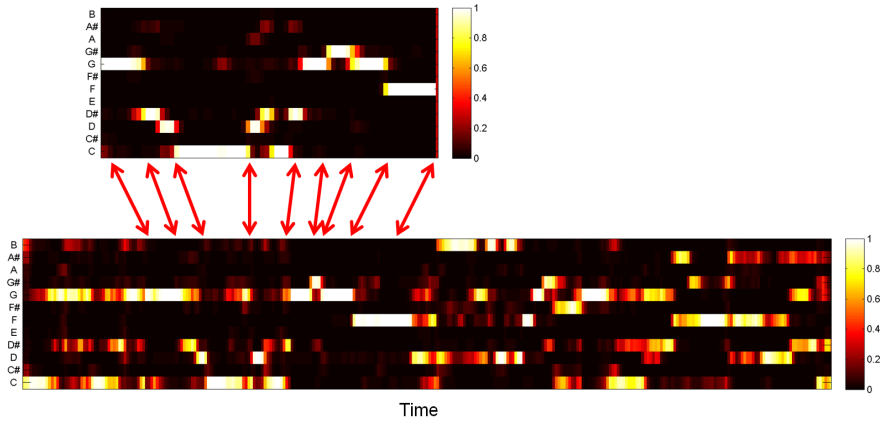

Figure 4 Chroma-based audio matching procedure. The red arrows indicate temporal correspondences between the query clip and a local section of a given music recording.

identifying the fragment (i. e., retrieving the audio recording containing the fragment along with the fragment's position) within a large audio collection [1, 5, 46, 68]. Note that at this level, the notion of similarity is rather close to the identity. Even though recent identification algorithms show a significant degree of robustness towards noise, MP3 compression artifacts, and uniform temporal distortions, existing algorithms for audio identification can not deal with strong non-linear temporal distortions or with other musically motivated variations that concern, for example, the articulation or instrumentation.

In sound track generation scenarios as described in [69], where the goal is to identify possible transition points within the same music recording, such strict notions of similarity may be meaningful. However, when changing from one music recording to a completely different one, a much coarser notion of similarity to identify potential transition regions is required. The identification of such regions can be accomplished by using *audio matching* techniques, where the goal is to retrieve all audio clips that musically correspond to the query [54, 50, 45]. In audio matching, opposed to traditional audio identification, one allows variations in musical aspects such as tempo, instrumentation, loudness, timbre, or accentuation.

In our proposed pipeline, we are specifically looking for audio clips that are harmonically related. Therefore, we use a chroma-based audio matching procedure as originally described in [54]. The general idea is to convert the audio material into mid-level representations that show a high degree of robustness to variations that are to be left unconsidered in the comparison. On the other hand, the feature representations should capture characteristic information that musically relate the identified clips to facilitate a plausible transition. In this context, *chroma-based audio features* have turned out to be a suitable mid-level representation [3, 24, 51]. 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. Representing the short-time content of a music representation in each of the 12 pitch classes, chroma features[1] show a large degree of robustness to variations in timbre and dynamics, while keeping sufficient information to characterize the rough harmonic

[1] MATLAB implementations for some chroma variants are supplied by the Chroma Toolbox: `www.mpi-inf.mpg.de/resources/MIR/chromatoolbox`, see also [53]

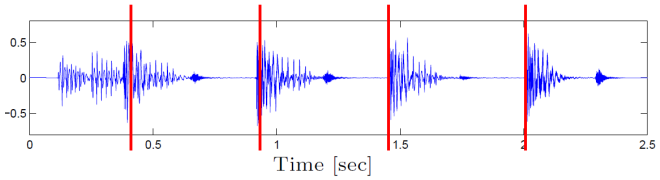

Figure 5 Beat tracking result (indicated by the red vertical lines) for a given music recording.

progression of the underlying piece of music. Based on these feature representations, the query clip is locally compared with clips that are contained in the target music recordings using alignment techniques. In particular, we use a local variant of *Dynamic Time Warping* (DTW) that can be used to find optimal temporal correspondences between the query clip and a local section of a given music recording [51]. Intuitively, these correspondences can be thought of a linking structure as indicated by the red arrows shown in Figure 4. These arrows encode how the feature sequences are to be warped (in a non-linear fashion) to match each other.

In [54], the main application of audio matching is to identify different versions of the same piece of music irrespective of the performance, instrumentation, or arrangement. As reported in [45, 29], using a query length of roughly 20 seconds (or more) leads to a high precision for this task. Now, in the sound track generation scenario as tackled in this paper, one is typically not interested in different versions of the same piece of music, but in harmonically related passages contained in *different* pieces. Such passages can be obtained when using query clips of shorter duration (less than 10 seconds). In other words, what is considered a false positive match in [54], may be a desirable match in our scenario.

3.3 Tempo and Beat Tracking

The chroma-based audio matching procedure is used to identify a target audio clip that shares a similar harmonic progression with the query clip. In view of a rhythmically plausible transition, one also needs to temporally synchronize the two clips—similar to what a DJ is doing when matching the beats of two recordings. This leads us to further central tasks referred to as *tempo estimation* and *beat tracking*, where the objective is to automatically locate the beat positions within a given music recording, see Figure 5.

Most approaches to tempo estimation and beat tracking proceed in two steps. In the first step, positions of note onsets within the music signal are estimated. Here, most approaches capture changes of the signal's energy or spectrum and derive a so-called *novelty curve*. The peaks of such a curve yield good indicators for note onset candidates [4, 9]. In the second step, the novelty curve is analyzed to detect reoccurring patterns and quasi-periodic pulse trains [12, 17, 28, 57, 60, 72].

Even though most humans are able to tap along the musical beat when listening to a piece of music, transferring this cognitive process into an automated system that reliably works for a large variety of musical styles is a challenging task. In particular, beat tracking becomes hard in the case that a music recording reveals significant tempo changes. This typically occurs in expressive performances of classical music as a result of ritardandi, accelerandi, fermatas, and rubato [30]. Furthermore, the extraction problem is complicated by the fact that the notions of tempo and beat may not be clearly defined due to a complex hierarchical structure of the rhythm [56]. In particular, there are various levels that are presumed to contribute to the human perception of tempo and beat. All these difficulties and ambiguities

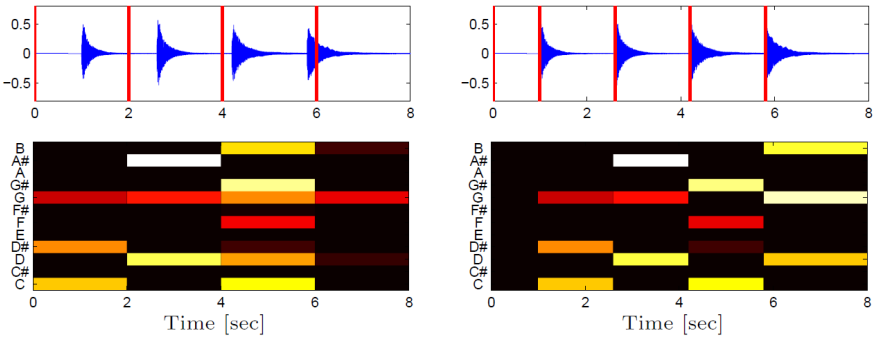

Figure 6 Waveforms and chroma representations using a fixed-size windowing strategy (left) and an adaptive windowing strategy using beat-synchronized windows (right).

have to be kept in mind when using the beat tracking results obtained from automated methods.

Knowing beat positions is not only necessary to temporally synchronize the query and target clip, as will be explained in Section 3.4, but is also beneficial for the feature computation and matching step as we now explain in more detail. When transforming a waveform into some feature representation, one typically splits up the signal into frames using a window function of fixed size and then applies the transform to each frame. Each feature value represents a local property averaged over the respective time window, which may result in "noisy" features when the signal's changes occur within a given window. As an alternative to fixed-size windowing, one can employ a musically more meaningful adaptive windowing strategy, where window boundaries are induced by previously extracted onset and beat positions. Since musical changes typically occur at onset positions, this often leads to an increased homogeneity within the adaptively determined frames which often improves the resulting feature representation, see Figure 6 for an illustration. One major advantage of using *beat-synchronized* audio features is that tempo differences between musically related audio clips are compensated [18]. This alleviates the requirement of using cost-intensive alignment procedures in the retrieval step as discussed in Section 3.2. Furthermore, knowing the beat positions allows for converting a physical time axis (given in seconds) into a musically meaningful time axis (given in beats or measures), which has huge benefits for presenting and comparing music analysis results [43]. However, when relying on beat-synchronous features, one should keep in mind that the quality of automatically extracted beats may be rather poor for certain types of music [30].

3.4 Time-Scale Modification

Once the beat positions are known in the query and target clip, one needs techniques that allow for locally speeding up or slowing down a music recording without changing other characteristics such as the pitch. Originally introduced for speech signals, there are numerous time stretching or *time-scale modification* (TSM) procedures. Most of these procedures are based on a fundamental technique referred to as *Overlap-and-Add* (OLA). The idea is to generate local copies of audio segments, which are obtained by windowing the original audio signal using suitably shifted Hann windows. These copies are then added up (using a constant window overlap) to produce the time-scale modified signal, see Figure 7 for an illustration. Generally, this simple procedure often results in severe noise-like phase distortions and stuttering artifacts which strongly downgrade the quality of the music signal.

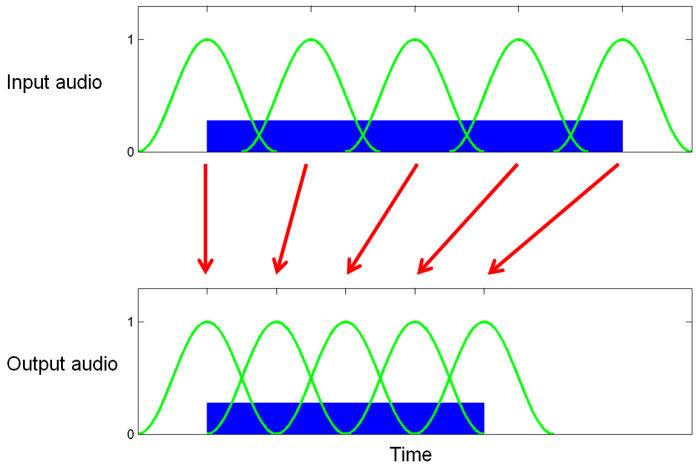

Figure 7 Illustration of the Overlap-and-Add (OLA) technique with blue indicating the waveforms and green indicating the windows.

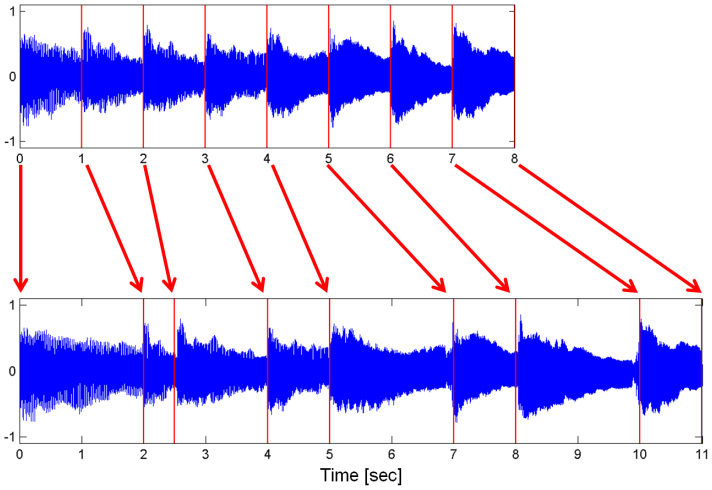

Figure 8 Non-linear time-scale modification of a music recording to temporally adjust beat positions.

Various time-scale modification algorithms have been proposed that try to attenuate these distortions. In general, one can distinguish between time-domain and frequency-domain approaches. A widely used time-domain procedure is known as *WSOLA* (waveform-similarity-based overlap-add) algorithm [65]. Here, phase discontinuities in the fundamental frequency are prevented by slightly adapting the window positions to obtain the local copies using correlation measures before the accumulation step is applied. On the other side, the most common frequency-domain approach is known as *phase vocoder* [13], where one first generates local copies as in the OLA procedure. Next, the phases of each local copy are adjusted in the Fourier domain to achieve a frequency-wise phase coherence in the subsequent accumulation step. To cope with various kinds of artifacts, numerous variants and hybrid methods have been proposed, see, e. g., [2, 14, 15, 25, 27, 40].

In our sound track generation scenario it is of particular importance that the used TSM procedure is capable of performing non-linear time-scale modifications. This is for example

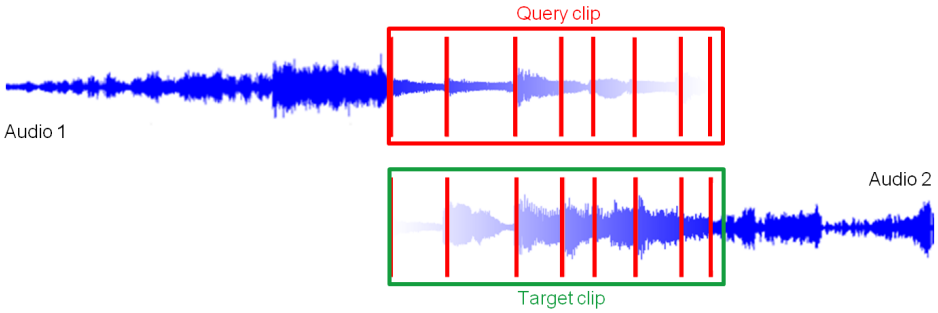

Figure 9 Cross-fade between beat-synchronized query clip and target clip.

needed when adjusting the beat grid of a music recording as shown in Figure 8. Finally, we note that the problem of pitch shifting with the objective to change the pitch of an input signal without changing its duration is dual to the time stretching problem. Here, to shift the pitch of a signal, one can first apply a time-scale modification procedure to stretch the signal and then use a simple sample rate conversion.

3.5 Intelligent Equalization and Blending

After the harmonically related query and target clips have been rhythmically synchronized, one can compute a transition by applying a simple cross-fade between these two clips. Then the transit from the current recording to the subsequent recording can be accomplished smoothly using this transition, see Figure 9.

So far, harmonic and rhythmic aspects were used for retrieving and adjusting the query and target clips. There are many more musical aspects such as instrumentation, musical voices or melodic structures one may want to consider in the transition. To this end, one requires techniques that allow for manipulating the audio material with regard to such aspects. This leads us to another fundamental and challenging area of signal processing generally referred to as *source separation*, where the goal is to decompose a given mixed audio signal into its individual sound sources.

In the musical context, source separation often deals with automatically extracting individual tracks that correspond to different instruments or musical voices from a given audio recording, see [10, 52, 67] for an overview. A related task is to parameterize an audio recording of a piece of music, where the parameters encode musical aspects such as pitch, onset positions, note durations, as well as tuning and timbre aspects corresponding to specific instruments [33, 48]. Exploiting the availability of additional information such as musical scores, various score-informed source separation strategies have been proposed [19, 20, 21, 31, 71]. Having an explicit control over the various sources allows for building musically meaning equalizers (instead of simple frequency-based equalizer) that allow for amplifying or attenuating certain voices (instead of frequency bands), see, e. g., [38, 42] and Figure 10.

Decomposing a monaural audio signal into musical voices is, in general, an extremely difficult problem. A special case is the decomposition of a music signal into a harmonic and a percussive component. Here, various methods have been proposed based on matrix decompositions of a spectrogram representation using machine learning techniques [34, 23, 47, 64]. In [55], a simple and fast algorithm that does not require any training material is proposed. This iterative approach relies on the assumption that harmonic components correspond to horizontal and percussive components to vertical structures within a spectrogram.

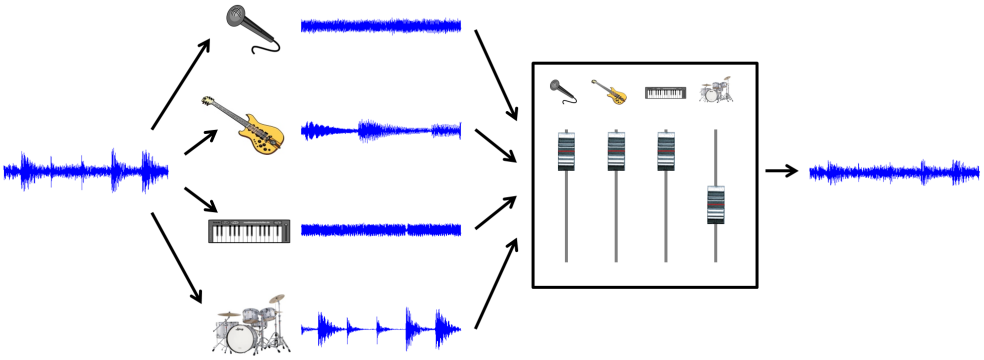

Figure 10 Instrument-wise equalization of a music recording (similar to [38, 42]).

In view of a sound track generation scenario, source separation and voice equalization techniques are important building blocks for the blending and morphing stage. Here, for example, one may want to suppress distracting voices or to amplify percussive components while concealing harmonic inconsistency. Actually, such techniques are also applied by DJs, who often amplify low-frequency bands while attenuating disturbing high-frequency bands in transition regions.

3.6 Indexing and Data Structures

In view of online capability of an overall sound track generation system, the *efficient* identification of suitable transition regions becomes an important issue. In the following, we want to touch on indexing and data structure issues.

Various indexing techniques have been applied for content-based audio retrieval. In case of audio identification, standard hashing techniques can be applied to obtain very efficient systems, see, e. g., [68]. For retrieval tasks on a lower specificity level, indexing become much harder because the *temporal order* of events, as also emphasized in [7], is of crucial importance for building up musically meaningful entities such as melodies or harmonic progressions. To account for temporal context, one often reverts to small chunks of audio also referred to as *audio shingles*, which leads, however, to features of high dimensionality. To index such high-dimensional shingles, techniques such as local sensitive hashing (LSH) have been applied for tasks such as *cover song identification* [6]. Here, being a document-based retrieval scenario, a bag-of-feature approach is applied with the features being the audio shingles. Such bag-of-feature approaches are not directly applicable to fragment-based retrieval scenarios such as audio matching. In [45], an indexing method is described based on inverted files which, however, only scales to medium size datasets. The idea of applying shingling and LSH-based indexing techniques to audio matching, where a single shingle corresponds to an entire audio clip of 10 to 20 seconds of duration, is investigated in [29].

Another idea to speed up the identification of transition candidates is to build up a graph-like data structure that explicitly encodes musical relations between audio clips. Such a data structure can be constructed from the given audio database in an off-line preprocessing step. As starting point, we want to take up an idea from the field of computer animation. Here, analogous to our music scenario, one important task consists in creating realistic, controllable motions from prerecorded motion capture sequences. In [44], a procedure is presented where a directed graph, referred to as *motion graph*, is constructed from a given

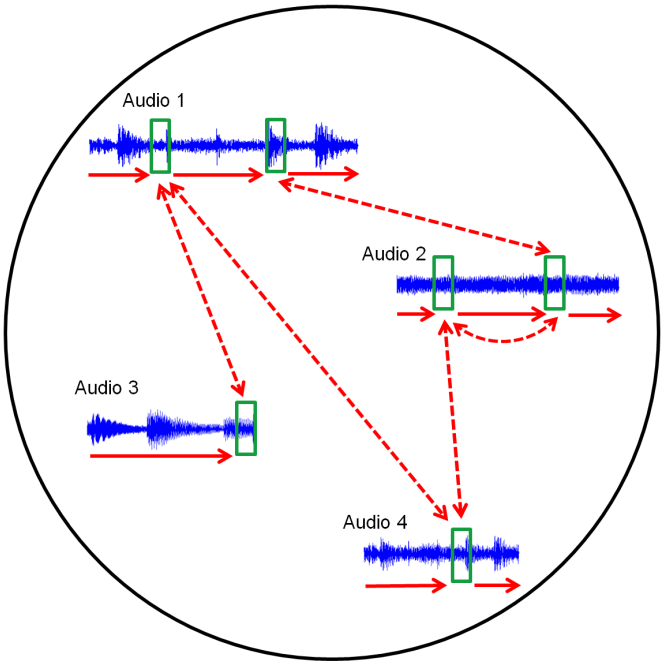

Figure 11 Music graph in analogy to the motion graph introduced in [44].

corpus of motion capture data. The edges of the graph contain either pieces of original motion data or automatically generated transitions, and the nodes serve as choice points where these small bits of motion join seamlessly. Motions can then be generated simply by building walks on the graph. Figure 11 illustrates this idea transferred to the music domain, see also [35] for a similar concept.

4 Conclusions and Future Work

The main goal of this contribution was to show how different aspects of music retrieval and audio processing come into play when dealing with applications such as data-driven sound track generation. Rather than presenting a concrete system, we sketched a possible pipeline for an online approach while discussing the necessary "ingredients" such as category-based music classification, content-based audio retrieval, beat tracking, time-scale modification, instrument equalization, and audio indexing. The intertwining and interaction of the various tasks is again summarized and illustrated by Figure 12. Each of the mentioned tasks constitutes itself a challenging research area with many open issues, in particular when dealing with various genre and styles of music—we have given numerous pointers to the literature that represent the state-of-the-art for the respective tasks.

Of course, when it comes to an actual realization and implementation of a concrete sound track generation system, many more challenges arise and a complete automatization of all steps neither seems feasible nor meaningful. However, there are many variants and more restricted sound track generation scenarios that come into reach. One such scenario is described in [69], where the duration of a given music recording is to be adjusted by suitably deleting, copying, and rearranging certain parts of the recording while keeping the flow of the music. Extending this scenario, a user may want to add background music to a slide show, where he specifies for each slide a desired music recording as well as a duration parameter.

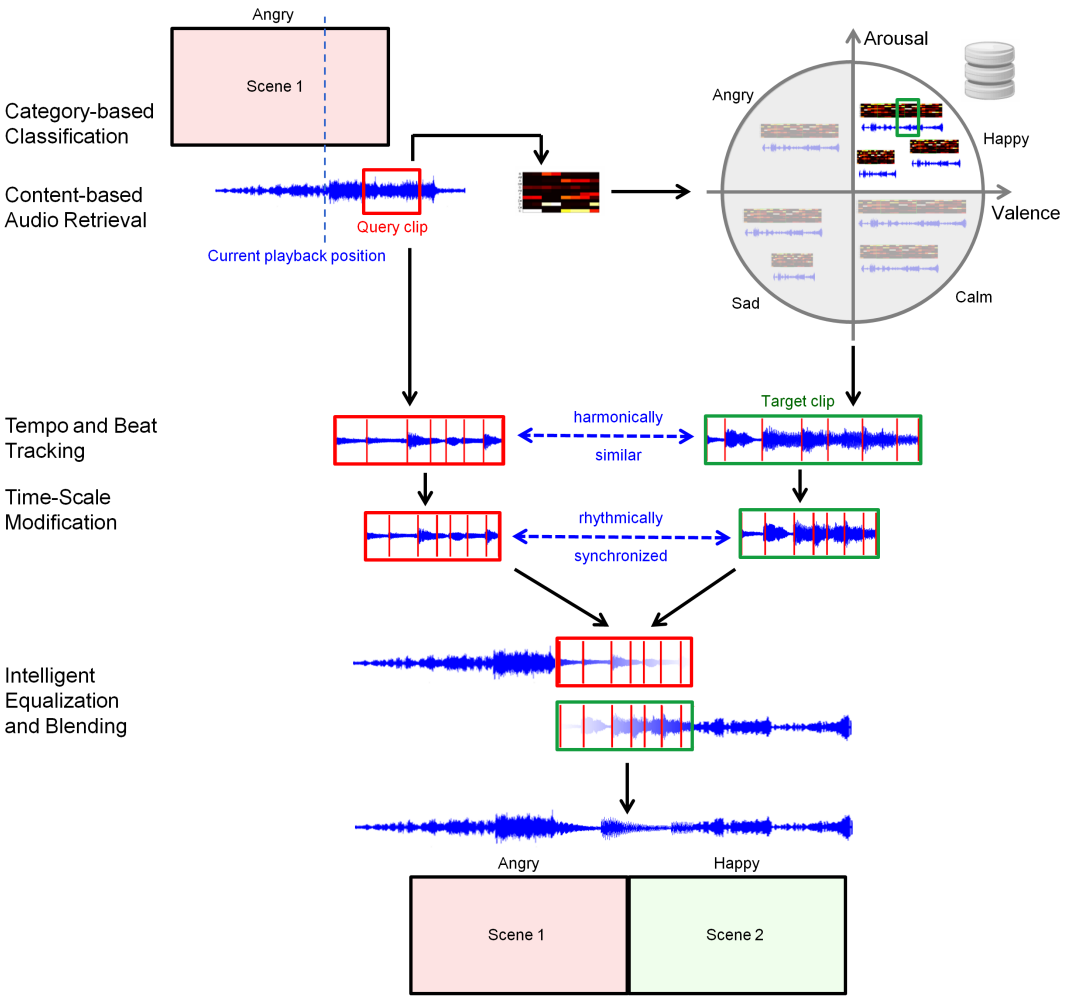

Figure 12 Possible pipeline for an automated sound track generation system.

Then, the task would be to automatically find and reassemble suitable parts of the recordings that not only fulfill the user constraints but also allow for euphonious transitions. Here, when the slide show is known in advance, an offline optimization procedure may be acceptable and efficiency issues become less significant. Furthermore, there may be different types of transitions a user may be interested in. For example, if there is a sudden event in the visual data stream, one may also want to have a surprising element in the sound track. Here, an abrupt change from one music clip to another may be acceptable or even desired. Instead of "complete solutions" that have been computed in a fully automated fashion, a user may rather need flexible tools that allow him to identify, modify, and assemble audio material in an intuitive and interactive way. Finally, perceptual issues need to be taken into account when it comes to the final assessment of the generated sound track. This itself constitutes an extremely difficult and interdisciplinary research area.

We hope that with this contribution we not only have given a useful overview of various tasks indicating challenges and future research directions, but could also give the reader an impression of the richness, depth and relevance of the research conducted in fields of music information retrieval and music processing.

5 Acknowledgment

This work has been supported by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI) and the German Research Foundation (DFG MU 2682/5-1). We would like to express our gratitude to Frank Kurth and Hiromasa Fujihara 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 Dan Barry, David Dorran, and Eugene Coyle. Time and pitch scale modification: A real-time framework and tutorial. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, 9 2008.

3 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.

4 Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

5 Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of algorithms for audio fingerprinting. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 169–173, St. Thomas, Virgin Islands, USA, 2002.

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

7 Michael Casey and Malcolm Slaney. The importance of sequences in musical similarity. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, 2006.

8 Dave Cliff. Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks. Technical report, HP Laboratories Bristol, 2000.

9 Nick Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *AES Convention 118*, Barcelona, Spain, 2005.

10 Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press, Elsevier, 2010.

11 David Cope. *Experiments in Musical Intelligence*. A-R Editions, Inc., 1996.

12 Matthew E.P. Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

13 Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.

14 Mark Dolson and Jean Laroche. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, 1999.

15 David Dorran, Eugene Coyle, and Robert Lawlor. Audio time-scale modification using a hybrid time-frequency domain approach. In *Proceedings Workshop on Applications of Signal Processing (WASPAA)*, New Paltz, New York, USA, oct 2005.

16 J. S. Downie. The music information retrieval evaluation exchange (2005–2007): a window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

17 Daniel P.W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

18 Daniel P.W. Ellis, Courtenay V. Cotton, and Michael I. Mandel. Cross-correlation of beat-synchronous representations for music similarity. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 57–60, Taipei, Taiwan, 2008.

19 Sebastian Ewert and Meinard Müller. Score-informed voice separation for piano recordings. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 245–250, Miami, USA, 2011.

20 Sebastian Ewert and Meinard Müller. Using score-informed constraints for NMF-based source separation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

21 Joachim Ganseman, Paul Scheunders, Gautham J. Mysore, and Jonathan S. Abel. Source separation by score synthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, New York, USA, 2010.

22 Loy Gareth. *Musimathics: The Mathematical Foundations of Music, Volume 1.* The MIT Press, 2006.

23 Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 16(3):529–540, March 2008.

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

25 Philippe Gournay, Roch Lefebvre, and Patrick-Andre Savard. Hybrid time-scale modification of audio. In *Audio Engineering Society Convention 122*, 5 2007.

26 F. 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. Available online: http://mtg.upf.edu/node/440.

27 Shahaf Grofit and Yizhar Lavner. Time-scale modification of audio signals using enhanced wsola with management of transients. *IEEE Transactions on Audio, Speech & Language Processing*, 16(1):106–115, 2008.

28 Peter Grosche and Meinard Müller. Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.

29 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)*, Kyoto, Japan, 2012.

30 Peter Grosche, Meinard Müller, and Craig Stuart Sapp. What makes beat tracking difficult? A case study on Chopin Mazurkas. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 649–654, Utrecht, The Netherlands, 2010.

31 Yushen Han and Christopher Raphael. Desoloing monaural audio using mixture models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 145–148, Vienna, Austria, 2007.

32 Gerhard Haupenthal. *Geschichte der Würfelmusik in Beispielen.* PhD thesis, Universität des Saarlandes, 1994.

33 Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 327–332, Kobe, Japan, 2009.

34 Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. EUSIPCO*, September 2005.

35 Alexander Höck, Frank Kurth, and Michael Clausen. Eine graphbasierte Indexstruktur zum inhaltsbasierten Audioretrieval. In *Proceedings of the Deutsche Jahrestagung für Akustik (DAGA)*, pages 185–186, Berlin, Germany, 2010.

36 X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. Ehmann. The 2007 mirex audio mood classification task: lessons learned. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, 2008.

37 Hiromi Ishizaki, Keiichiro Hoashi, and Yasuhiro Takishima. Full-automatic dj mixing system with optimal tempo adjustment based on measurement function of user discomfort. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–140, Kobe, Japan, 2009.

38 Katsutoshi Itoyama, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models. In *Proceedings of the International Conference for Music Information Retrieval (ISMIR)*, pages 133–138, Philadelphia, USA, 2008.

39 Tristan Jehan. *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, 2005.

40 Nicolas Juillerat, Stefan Mueller Arisona, and Simon Schubiger-Banz. A hybrid time and frequency domain audio pitch shifting algorithm. In *Audio Engineering Society Convention 125*, 10 2008.

41 Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull. Music emotion recognition: a state-of-the-art review. In *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pages 255–266, 2010.

42 Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Musical instrument recognizer "instrogram" and its application to music retrieval based on instrument similarity. In *IEEE international symposium on multimedia*, pages 265–272, San Diego, California, 2006.

43 Verena Konz, Meinard Müller, and Sebastian Ewert. A multi-perspective evaluation framework for chord recognition. In *Proceedings of the International Society for Music Information Retrieval Conference(ISMIR)*, pages 9–14, Utrecht, The Netherlands, 2010.

44 Lucas Kovar, Michael Gleicher, and Fréd éric Pighin. Motion graphs. *ACM Trans. Graph.*, 21(3):473–482, 2002.

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

46 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.

47 D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS*, pages 556–562, 2000.

48 Pierre Leveau, Emmanuel Vincent, Gaël Richard, and Laurent Daudet. Instrument-specific harmonic atoms for mid-level music representation. *IEEE Trans. Audio, Speech and Language Processing*, 16(1):116–128, 2008.

49 Heng-Li Lin, Yin-Tzu Lin, Ming-Chun Tien, and Ja-Ling Wu. Music paste: Concatenating music clips based on chroma and rythm features. In *10th International Society for Music Information Retreival Conference (ISMIR 2009)*, pages 213–218, 2009.

50 Riccardo Miotto and Nicola Orio. Automatic identification of music works through audio matching. In *ECDL*, pages 124–135, 2007.

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

52 Meinard Müller, Daniel P. W. Ellis, Anssi Klapuri, and Gaël Richard. Signal processing for music analysis. *IEEE Journal on Selected Topics in Signal Processing*, 5(6):1088–1110, 2011.

53 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.

54 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.

55 Nobutaka Ono, Kenichi Miyamoto, Hirokazu Kameoka, and Shigeki Sagayama. A real-time equalizer of harmonic and percussive components in music signals. In *Proc. ISMIR*, pages 139–144, Sepember 2008.

56 Richard Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11:409–464, 1994.

57 Geoffroy Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Advances in Signal Processing*, 2007(1):158–158, 2007.

58 J. A. Russell. A circumspect model of affect. *Journal of Psychology and Social Psychology*, 39(6):1161, 1980.

59 N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.

60 Eric D. Scheirer. Tempo and beat analysis of acoustical musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

61 Björn Schuller, Florian Eyben, and Gerhard Rigoll. Fast and robust meter and tempo recognition for the automatic discrimination of ballroom dance styles. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 217–220, 2007.

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

63 G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing*, 5(10):293–302, 2002.

64 Christian Uhle, Christian Dittmar, and Thomas Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proc. ICA*, pages 843–847, April 2003.

65 Werner Verhelst and Marc Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Minneapolis, USA, 1993.

66 David Vink. Adaptive music for video games. `http://www.gamecareerguide.com/features/768/student_thesis_adaptive_music_for_.php` (retrieved 19.02.2012), 2009.

67 Tuomas Virtanen. Unsupervised learning methods for source separation in monaural music signals. In Anssi P. Klapuri and Manuel Davy, editors, *Signal Processing Methods for Music Transcription*, chapter 6, pages 267–296. Springer US, 2006.

68 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.

69 Stephan Wenger and Marcus Magnor. Constrained example-based audio synthesis. In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo (ICME 2011)*, Barcelona, Spain, July 2011.

70 Guy Whitmore. Design with music in mind: A guide to adaptive audio for game designers. `http://www.gamasutra.com/resource_guide/20030528/whitmore_01.shtml` (retrieved 19.02.2012), 2003.

71 John Woodruff, Bryan Pardo, and Roger B. Dannenberg. Remixing stereo music with score-informed source separation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 314–319, 2006.

72 Ruohua Zhou, Marco Mattavelli, and Giorgio Zoia. Music onset detection based on resonator time frequency image. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1685–1695, 2008.