Jan-Philipp Tauscher

Supervisor Stephan Wenger

Referee Prof. Dr. Ing. Marcus A. Magnor

Co-Referee Prof. Dr. Ing. Friedrich M. Wahl

Audio Resynthesis on the Dancefloor: A Music Structural Approach

Audioresynthese für die Tanzfläche: Ein musikstruktureller Ansatz

Bachelor Thesis

July 23, 2012

Computer Graphics Lab, TU Braunschweig

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Braunschweig, 23. Juli 2012

Jan-Philipp Tauscher

Zusammenfassung

Im Rahmen dieser Bachelorarbeit soll ausgehend von der Grundidee eines bestehenden Verfahrens zur Audio Resynthese [WM11] ein neuer Ansatz für strukturierte Tonquellen entwickelt und neue Anwendungsfelder dafür erschlossen werden. Das bestehende Verfahren findet mögliche Sprungstellen in einem Musikstück und synthetisiert durch Aneinanderreihung von Segmenten des Stückes ein neues, das vorgegebene Randbedingungen erfüllt. Im Rahmen der Arbeit soll mittels Beat Tracking ein zuverlässigeres Matching rhythmischer Strukturen bei der Sprungstellensuche erreicht werden. Die Qualität der Ergebnisse soll mit Beispielen und Vergleichen aus dem Gebiet der kontemporären Tanzmusik und Gegenüberstellung mit dem ursprünglichen Verfahren demonstriert werden.

Weiterhin sollen algorithmische Verbesserungen ausgearbeitet werden, die das neu Zusammensetzen an nicht hörbaren Sprungpositionen nach gewünschter Ausgabestruktur ermöglichen, um im speziellen die einfache Erstellung von rearrangierten Stücken für einen Tanzchoreographen zu ermöglichen.

Abstract

This thesis improves and extends existing methods in the research area of audio resynthesis and retargeting and extends its usage scopes. The existing approach analyzes a musical piece for possible cut points that allow the resynthesis of a novel soundtrack by lining up the source segments according to specified rules. For the improvement of matching harmonic and rhythmic structures during cut points search, beat tracking is used as core component of this work. Segment rearrangement is improved by employing faster and better suited algorithms.

Contents

Chapter 1

Introduction

Music (Greek: mousike; "art of the Muses") is the science and art of arranging sound events and silence in temporal succession to create a continuous, consolidated, expressive composition, playing an important role in western societies. Dance as another major form of performing arts is usually intrinsically linked to a certain large subset of musical arrangements employing a regular, uniform structure, not taking into account experimental dance or silent dance like performances by auditory disabled people only relying on mechanically perceivable cues like bass tones. After initial composition of a musical score by an artist, the audible material is used and replayed in a variety of locations and social settings, raising the demand of editing and rearranging the record in the aftermath. Utilizing the approach of example-based audio synthesis we propose a method for completely automatic or user-supported rearrangement of regularly structured musical compositions with respect to usability in dance related institutions such as academies, theatres or clubs.

Especially in dancing academies and theatres, customly cut music is a core element of dance choreography design. Often, choreographers create a choreography according to a given song, limiting the creative dimension to that specific song arrangement. They usually have no sound engineering qualifications and need systems to support them during soundtrack creation. They would want to segment a song, find similar parts to jump between them without notice, rearrange them, cut them, extend them, stretch or shorten a song or one of its parts to a given duration, create a medley song from parts of different songs, insert measured aligned silence and so on.

We focus on music in the form of strictly structured audio, leaving out sound as audible textured patterns lacking defined pitch, rhythm, dynamics and timbral sonic qualities. This form of music plays a predominant role in almost all forms of societies today but differs in form, function and style throughout different cultures. Music drives and incarnates dance as a series of movements and steps performed to it, creates and manipulates emotions and mental states of the auditor and is subject to psychological immersion. Motion pictures without music would not be perceived in the way modern film-making creates them, as these human senses are associated closely. The same is true for electronic games and much work has been done in this field in the past decades [Col08]. Not only being an art itself, music supports and enhances the experience of artistic installations and exhibitions, though the evolutionary creativity and artistic ability is not yet easily mimicked by computers. [McC05] What we can already make computers and algorithms do, is analyze the structure of a musical piece to a certain extent. What we cannot do until today is teach a machine to reproduce the cognitive processes running in human minds, that correctly and completely analyse the underlying pattern of a musical performance, but the current technologies still provide us with the basic tools necessary for machine aided rework of a musical performance. This is where structurally aligned audible content as described above comes into the field that can be analyzed and segmented with prior-art technologies like beat trackers [Ros92] and structure analysis tools [Sch06] to be rearranged in the following process. This is particularly useful for dance applications where a choreographer can tailor a song to his specific demands [SA10] jumping back and forth in a track, shifting logical parts around, inserting silence or even parts from other songs like a medley (might involve time and pitch scaling), always being aligned to the beat grid. A dance hall DJ may automatically stretch a song to a desired length, play it forever by letting the computer randomly select the next playing position in the musical piece or jump to the end on key press within predefined time. The ability to do this seamless jump towards the outro part of a song might also be interesting in automotive applications when the final destination of a programmed satellite-guided route has been reached, the engines have been switched off or the driver's door has been opened. Another form of art, involving but not mutually exclusively demanding music, is performance juggling. In the undesirable case of the juggling equipment falling to the ground during a performance show involving musical choreography, the soundtrack needs to be resumed at a certain location after the performer has picked up his equipment and is ready to recover the show, while the sound played on uninterrupted up to this point.

The previously stated uses cases for motion pictures as in [WM11] still apply. Composing, performing and producing a soundtrack for movies, video clips, computer animation video games or similar visual content is a profoundly challenging, time intensive process. Employing an automatic system for scaling the soundtrack to the boundaries of a video source, with respect to some certain key points in it, would be highly desirable. Real-time soundtrack generation for electronic games' live event adaption is also enhanced by taking the underlying basic structure of the piece of music into account leading to a more seamless musical experience.

These applications share the similar requirement of synthesizing a soundtrack from a given piece of music towards the user supplied constraints. We modify the algorithm proposed by [WM11] to meet the constraints of dance compatible and music structure aware applications. First, a soundtrack is segmented into its elementary logical parts and its measures, using beat tracking. With a self-similarity analysis of this partition, we generate a jump-table containing appropriate transition points within the song for seamless jumps between different sections of the track corresponding to the users' constraints. This method is given impetus by cut-and-stitch approaches found in computer graphics [LHL10]. We evaluate different algorithms for optimizing the quality of these transition points. After musically correct cuts become available, we transform the use cases stated above into an application with real-world relevance, that is a software that simplifies some of the tasks encountered in music and dance related environments. At last, we try to improve the current art of music structure recognition, to enable the rearrangement of whole parts of a musical piece, either automatically as above, or according to user guided instructions and implement it in usable software.

Chapter 2

Related Work

The contemporary Concatenative Audio Resynthesis approach by cutting and stitching started in the 1950s but was not known as it is today until the past decade [LWZ04, PB04]. A good overview is given in [Sch06].

These methods generate a partition of an audio source and build a new audio piece of arbitrary length. The transition probabilities, along which reconstruction takes place do not catch the source structure and therefore let past approaches fail on more complex musical pieces.

Audio and User Directed Sound Synthesis calculates the self-similarity of an audio source per frame and has been successfully tested on stochastic or periodic sound pattern but not on music [CBR03]. Audio Textures by [LLW$^+$02] segments input audio into short clips for Mel analysis, similarity measurement with auto-correlation of temporal neighbours and sequence-decided recombination and or overlay with possible effects like pitch shifting, time scaling and amplitude setting to be added to avoid monotony but is only applicable to sound textures and not to music. Strobl et al. summarizes similar sound texture generation methods [SERlG06].

Concatenative Audio Synthesis [Sch06, S$^+$00] uses databases of sound snippets to assemble a specific target sound. While this approach has been used in electronic music composition through arranging small sound snippets like a mosaic [Stu04, ZP01], it mainly works for constructing sound textures and sound scapes and does not perform well on musical sources. A concatenative system analyses a sound source by segmenting and describing its characteristics and stores this information in a database. A target specification will be generated from analysis of another sound source as described or from a symbolic score of descriptors. Then, segments are selected from the database according to a distance and concatenation quality function to best match the given target specifications. Finally, selected segments are concatenated, either freely placed in time for speech synthesis or statically at defined positions for rhythmic synthesis [Sch05].

In the past, synthesis of sound textures and patterns has been researched to a great extent. These schemes usually fail on structured musical sources. To push the border on these limitations, [WM11] introduced a multi-resolution scheme for fast self-similarity analysis capturing the sonic source from large-scale structure down to single samples allowing perfect alignment of the cuts without blending or scaling. This approach performs reasonably well even on structured music but has limitations regarding the usability of the synthesized soundtracks in music theory aware environments. The jumps proposed by this system do not necessarily correspond to the underlying meter of the musical source, causing irritation to the musically trained listener or dancer who expects temporal continuity. Even the inexperienced listener is able to gain basic knowledge about a song's structure simply by applying life-long informal musical training [TP80] like listening to the radio. Another problem are jumps within the track occuring at positions that fit harmonically but not at its musical depth level, like solo vs. tutti instumentation, or do not match well at all.

In this thesis we are going to introduce a concept for resynthesis of structured musical sources with respect to its underlying structure. Here, Audio-based music structure analysis like [GM94, PMK10, Cha05] assistively drops in, aiming to segment the audio track into its logical pieces defined by the underlying beat, human perception and music theory. We can safely ignore the fact that this approach cannot cope with sound having no structure or a structure not supported by the employed analytic concept, because this would void the usability for dance and motion arts applications. This allows us to come up with a resynthesis solution, suitable for operation in music structure aware environments by piecing together a target song, using only elements structurally applicable like a measure.

Chapter 3

Beat Tracking

We describe our approach for the segmentation of music to find locations in it, allowing seamless transitions to other positions in this chapter. A listener should not notice these jumps in a song, at least not through rhythmic violations, that is the temporal misalignment of musical material. Recognizing the beat in a given piece of music is crucial for basic analysis. It provides us with necessary information to find cuts that will not void the usability as a dance track and to recognize logical entities in a song at measure level. Therefore we start out with the description of the science and art of beat tracking.

3.1 What is Beat?

The beat is the elementary time unit in music, its rhythmic pulse. Grouped beats form a measure and its number of notes and rests corresponds to the meter, also called time signature, usually contributed by percussion instruments. Simple time signatures are $\frac{4}{4}$, $\frac{3}{4}$, $\frac{6}{8}$, complex $\frac{5}{4}$, $\frac{7}{8}$, fractional $\frac{2\frac{1}{2}}{4}$, irrational [Fer] $\frac{3}{10}$, $\frac{5}{24}$. Music without any percussive instruments has an implicit beat noticeable through chord changes or its note alignment. A strong and easily recognizable beat is a necessity in popular music to attract the mass auditorium, unlike the absence of distinct beat patterns as in experimental or avant-garde modern music. Classical music is usually performed in a very expressive way, strongly varying the rhythm and beat, to convey tension and atmospheric manipulation the casual listener is often not familiar with. Most commonly known music attracts the audience with its fascinating, catchy beats but is very static in performance. Strong beat patterns of this kind enable even untrained listeners the possibility of tapping in time with the beat. In dance music, the time signature, that is the number of beats contained in one measure, does not change throughout the song to enable a continuous performance. Measures are usually grouped into larger, clearly audible entities, like verses and a chorus.

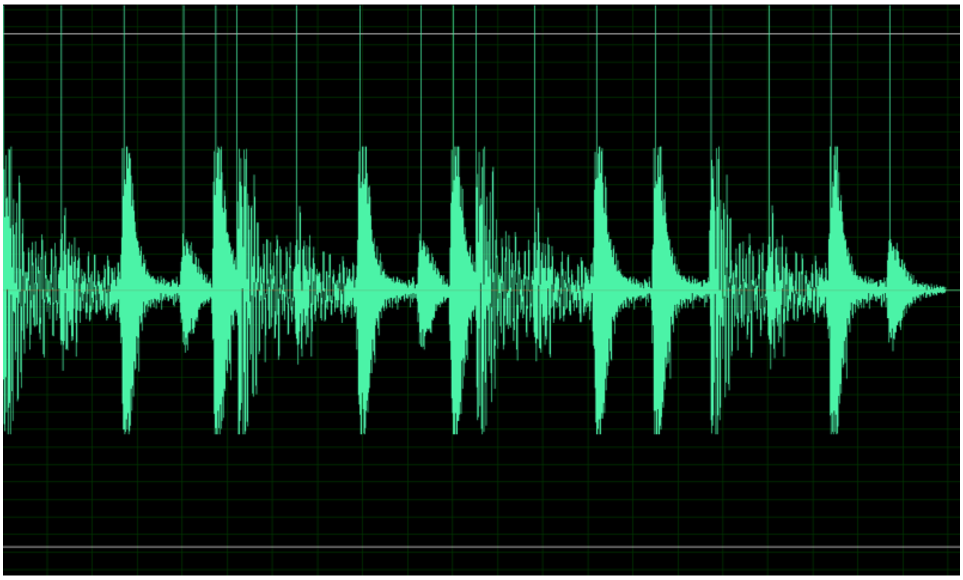


Figure 3.1: A sample drum loop with onsets marked [Flo11]

3.2 Tracking the Beat

The recognition of beat in music in usually divided into certain phases. The first one copes with the detection of note onsets in the song's signal (Fig. 3.1), this is computing a *novelty curve* for recording changes in energy, spectral content or pitch [BDA$^+$05, DP07, Ear07, KEA06]. The peaks of this curve represent the possible onsets and are chosen by some kind of selection mechanism [BDA$^+$05]. The next phase gives an estimation of the local tempo of the musical track by analysing its onset patterns for recurrence and periodicity [DP07, KEA06, Pee07]. The tempo is assumed to be constant within the local analysis window, marking the trade-off between tempo robustness and detection of tempo changes with respect to the window size. The last phase selects the appropriate sequential beat positions for a correct description of the piece's periodic beat structure (Fig. 3.2), regarding frequency of tempo and phase of timing.

3.3 Machine Perception Limitations

Even today, beat tracking is a challenging task. Humans are usually able to tap in time with the beat flawlessly. Over the past years, quite a number of algorithms has evolved, coping with the extraction of beat positions in music [GD05]. This process, as executed by a human's cognitive capabilities, is difficult to model as a machine-driven, automated solution. A beat is a perceptual phenomenon and does not necessarily correspond to physical beat

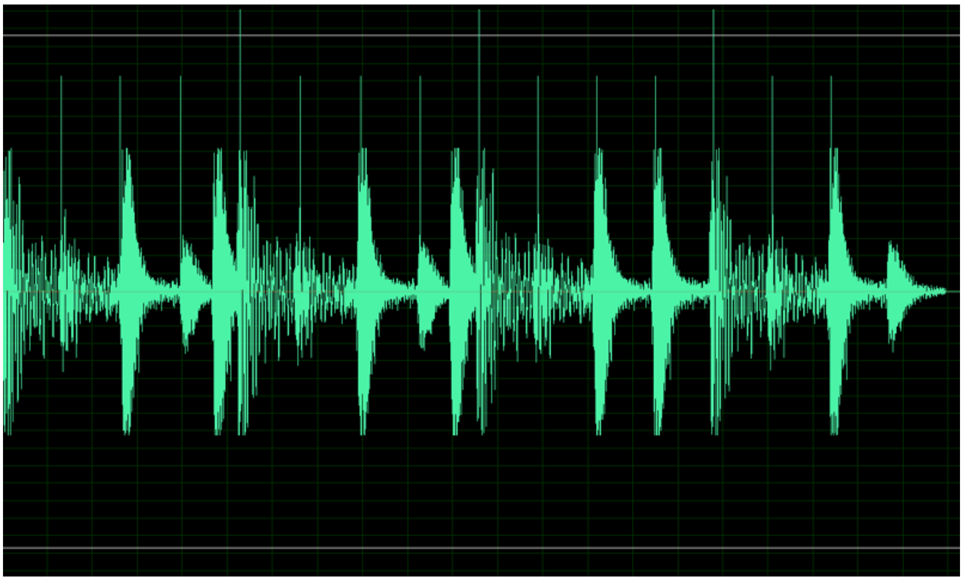


Figure 3.2: A sample drum loop with estimated onsets marked. Long lines show the down beats [Flo11]

times. It is usually accompanied by a note onset defined by strong energy in the time-domain of the signal or altered spectral content. These hints of a beat structure might be hidden behind soft note onsets, blurred note transitions or delayed beats, like off or back beats, leading to mis-perception of the machine. Variations in the tempo increase these issues, and the number of different instruments in popular music make the retrieval of the precise note onsets difficult. Not only physical reasons account for complications in beat recognition, as a number of musical ones also raise the bar. In passages without a physically perceivable beat, a human listener might still be able to determine a continuous beat in the absence of note events going along with the actual beat, whereas the machine will fail, as an automatic determination of physical beat positions is difficult in this case, especially during varying tempo without straight interpolation possibilities. Changes of the time signature during the performance of a musical piece also accounts for severe tracking problems. Even for music professionals it is not trivial to detect and describe an accurate change of the time signature. Simple peak-detection is not sufficient since energy peaks do not necessarily correspond to beats. [GMS10] shows the difficulties arising in beat recognition especially with expressive performances, like romantic piano music [Ear07], marking the bar and understanding of current beat trackers. Contemporary pop and rock music with a strong beat and constant tempo is handled quite well by many solutions. [Dix01, MMDK07] give an overview of empirical evaluations of several beat tracking approaches.

3.4 Real World Assumptions

According to Section 3.1, music used in dance related environments follows some assumptions like static time signatures, defined song building blocks and a steady tempo. As described in Section 3.3, state-of-the-art beat trackers are able to cope with this content and produce reasonable results for real-world applications. As we focus on these dance environments, we are in a good position to employ contemporary beat tracking mechanisms as a reliable (in most cases) source for our research.

3.5 Selecting a Tracker

Recognizing the beat in a given piece of music is crucial for basic analysis. It provides us with necessary information to find cuts that will not void the usability as a dance track and to recognize logical entities in a song at measure level. This directly implies the first demand among the following other *demands of a beat tracker.*

Analyse and record beat structure The ability to correctly follow the beat of a song, as a natural person would do by tapping in time with the rhythm up to eighth note level for a good resolution. A transcript of all recognized beats and its respective positions in the song has to be created.

Determine time signature The metered time of a song has to be estimated to group the beats into measures. It is sufficient to focus on simple time signatures as described in Section 3.1.

Find measure beginnings Find the downbeat positions for correct song segmentation. The down beat marks the beginning of a measure.

Adapt to varying tempo As some songs may vary in their tempo, even if unnoticed by the untrained listener, tempo variations have to be recognized and followed. Adaption to the tempo is crucial as the recorded beat grid would lose synchronization with the song over time.

Many available beat trackers fail on these demands. Most common problems are the missing ability of finding measure onsets, not correctly detecting measure beginnings or not accurately finding the beat onsets at all. Despite this fact, many beat trackers tend to generate a linearly spaced beat grid that does not adapt to tempo changes in a song. After evaluating the performance of various beat trackers, like *BeatRoot* [Dix07], *beatsync* a simple proof of concept tracker, *BTrack* [SDP09], *B-Keeper* [RP07] and the tracking components available for the *Sonic Visualizer*[1] software, we selected the *[aufTAKT]*[2] *tempo and beat tracking system* by *zplane.development*[3], a Berlin based company focusing on state-of-the-art music processing and analysis technology research. It was the only beat tracker tested, to reliably track the beat signal in musical sources of our research area and has also been field tested by a number of well-known music stage processing and editing software vendors. [aufTAKT] analyses the input signal for its note onsets by detecting new energy and frequency components and weights them according to their perceptual importance. A beat analysis module computes the actual beat position from the onset information, even if the onsets do not necessarily correspond to the physical beat locations. [aufTAKT] determines the time signature, finds the first down beat of a measure and adapts to varying tempo of a musical source.

Not being perfect, [aufTAKT] fails on material containing time signature changes, lacking regular beat patterns or featuring other experimental or uncommon musical properties. These are general problems in the research domain of beat tracking affecting every beat tracker (see Section 3.3). This is irrelevant according to Section 3.4, because this kind of music is not usable on the dance floor anyway.

[1] www.sonicvisualizer.org, www.vamp-plugins.org

[2] [aufTAKT] V3 tempo and beat tracking

[3] zplane.development / Katzbachstr. 21 / D-10965 Berlin, Germany, *www.zplane.de*

Chapter 4

Audio Resynthesis on the Measure Level

4.1 Cold Starting the Resynthesizer

Now we start out describing our first approach on structural aware resynthesis and the first results. We are building on the results by [WM11] using beat tracking technology kindly provided by *zplane.development*[1].

The existing framework provided by [WM11] consists of a base module capable of reading audio files and already computed pre-results from hard disk, an algorithm for finding cuts in an audio source and a collection of algorithms for searching a path through the source according to the user-specified constraints.

Chapter 2 summarized current approaches on audio resynthesis and describes some flaws in the synthesized target song regarding musical structure. So our initial focus lies on the improvement of the cut finding algorithm to respect the basic song structure. To address these problems, we are going to employ *zplane*'s [aufTAKT] technology for recognition of the beat grid in a musical signal. The beat grid computed by [aufTAKT] contains the sample-accurate beat positions of the input song and lets us compute its measure boundaries. With this structural information now being available, we can do block-wise self-similarity analysis of the input song. For similarity computation, the measures found above are used as the respective blocks and treated as follows.

The general concept is to divide an audio source into into a useful sequence of features $(x_1, x_2, ..., x_n)$ and compare its elements pairwise according to some distance function d and store the results in a self-similarity[2]

[1] further referred to as *zplane*

[2] The counterpart to self-similarity matrices are self-distance matrices. The latter describe the distances between analysed frames instead of their similarity. This difference is only a matter of a scale-directional point of view.

matrix $S(i,j) = d(x_i, x_j) : i,j \in 1,2,3,...,n$. The first to use this concept were [EKR87] for analysis of chaotic systems, while [Foo99] introduced it to the domain of music analysis for visualization of an audio recording's time structure.

The lengths of the measures naturally differ by a few samples and need to be equalized for comparison. This is done by scaling the measures to match the length of the measure with fewest samples using 3^{rd} *order spline interpolation*. Then we compute the amplitude spectra of the length-equalized measures with the *Fast Fourier Transform* [CT65]. Self-similarity computation is done via the *Bray-Curtis* dissimilarity [BC57],

$$d(u,v) = \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|} \quad , \tag{4.1}$$

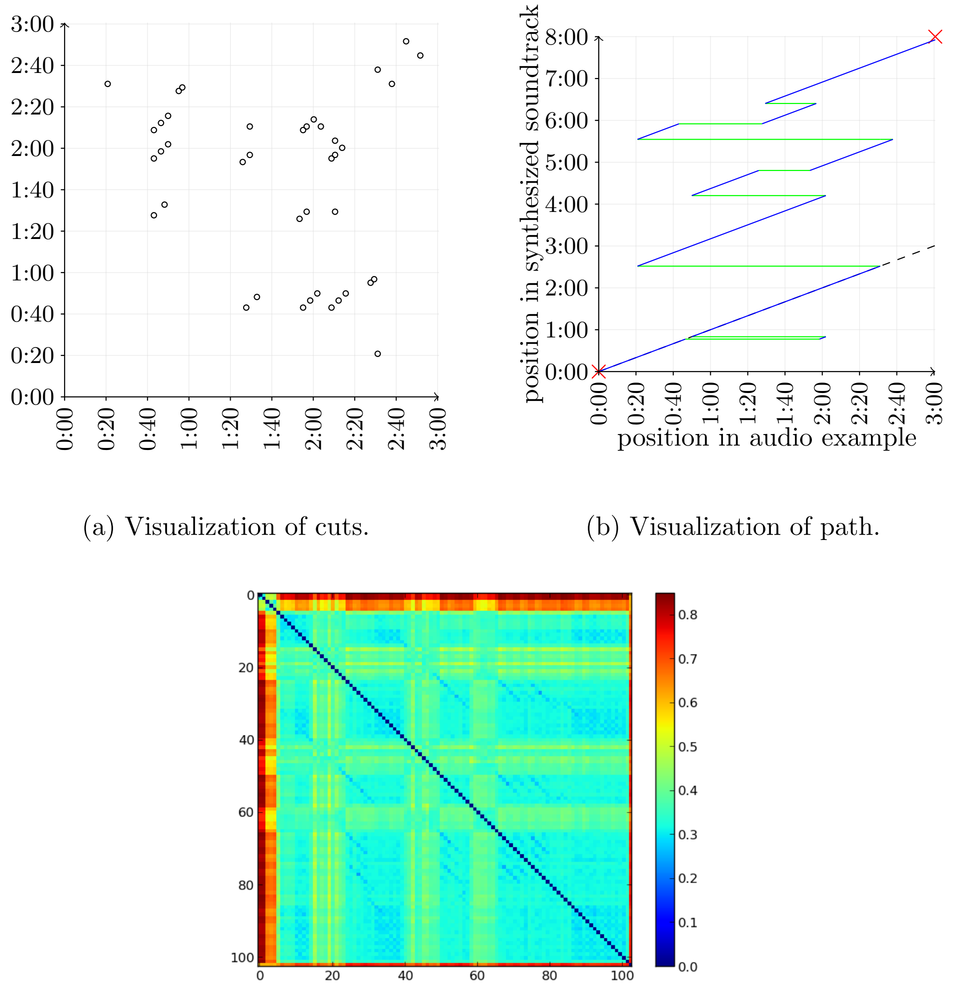
as error function and stored in a distance matrix (Figure 4.1c). It is one of the most well-known non-metric[3] ways of quantifying the difference between data sets and delivers robust and reliable dissimilarity results throughout many applications.

We keep the positions of the measures with lowest distances as cut points (see Figure 4.1a) for further processing. To avoid trivial cut points, the diagonal of the self-similarity matrix is ignored. The resulting cut positions are then fed to the *Genetic Path Algorithm* by [WM12]. This algorithm searches a path through the input song by constructing parts from the previously found cut positions according to the user-defined constraints like the final duration of the synthesized target song (Figure 4.1b).

4.1.1 Choke Results

Using the first approach as described above, we computed cuts directly sitting on the measure boundaries. The target song pieced together from these cut points contain many good sounding jumps in the songs that go unnoticed by the listener. Despite these good initial results, still a number of problems exist. We got a number of too short parts in the synthesized target, consisting only of one measure. Furthermore, there are cuts at the very beginning and end of a song to cause matching, but simple repetitions of the entire input song. Another problem is cuts that match harmonically but lack instrumental depth, that is a transition to a musically similar part that is performed with different instrumentation.

[3] It violates the last of three properties defining a metric: the triangle inequality

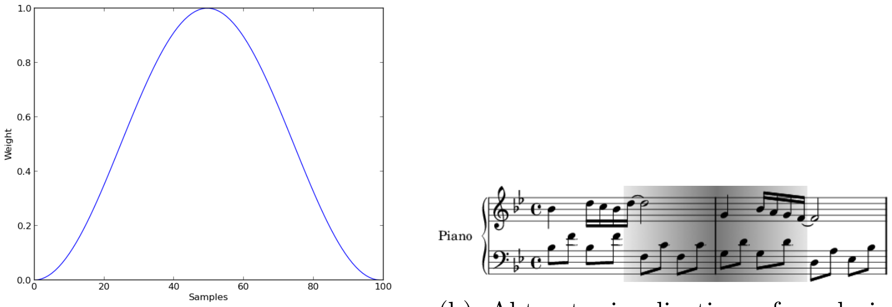


(a) Visualization of cuts.

(b) Visualization of path.

(c) Self-similarity matrix showing pair-wise similarity between measures.

Figure 4.1: Results of initial approach
Figure 4.1a shows locations in the song to jump perceptually smooth from the time position tagged on the abscissa to the time position shown on the ordinate. Figure 4.1b displays the path plot of desired length, computed using [WM11]'s genetic path algorithm, with the blue diagonals indicating the parts of the source song to be copied and the green lines visualizing the jumps in the source song. The self-similarity matrix is shown in Figure 4.1c. Blue colors indicate a high, red colors a low similarity. As every measure is very similar to itself, the diagonal is ignored during cut search.
This example shows *Deer in the Headlights* by *Adam Young*.



(a) Hanning window

(b) Abtract visualization of analysis window position

Figure 4.2: Analysis window and its position

- + cuts exactly at measure boundaries
- + many good sounding cuts
- − too short cuts, i.e. only one measure
- − cuts at the very beginning and end of a track causing simple repetitions
- − harmonically matching cuts but missing instrumental depth, that is a cut between similar parts but played by different instruments

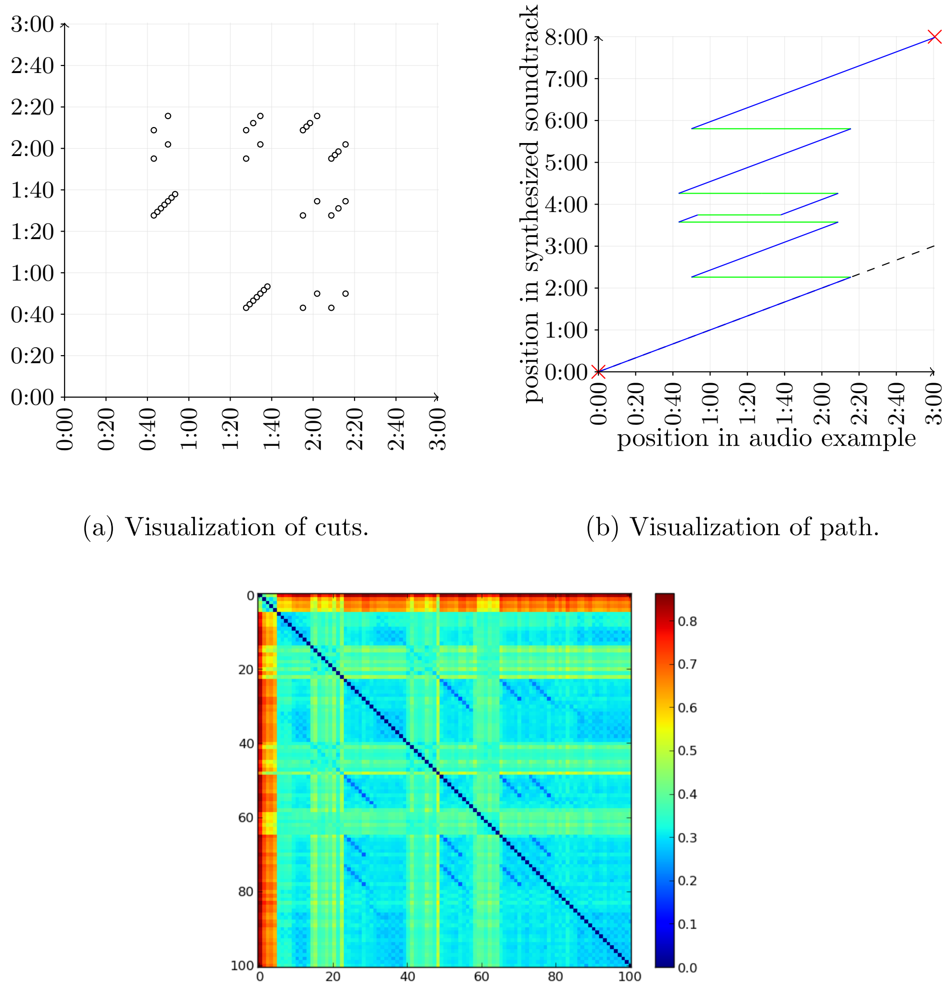
4.2 Refining the Comparison Metrics

Our initial approach resulted in many good sounding cuts but still has some flaws. For better transition results, the analysis focus is shifted from the single measure used as window before, to an area around the measure borders of the size of two times the mean size in samples of all measures in a song. We ignore the first and last measures to avoid problems with the analysis window size at the song borders and as a trivial solution for the repetition problem described in the previous section. The analysis window is then smoothed with a Hanning Window [BTTT59], a taper window commonly used in digital signal processing to weight the sampled input according to its definition, that fully includes the measure border and decays its inclusion to zero when approaching the window borders (Figure 4.2a).

4.2.1 Results of the Refined Comparison Metrics

The adjustment of the analysis window mainly resulted in harmonically better matching cuts. Due to the extended and weighted look[4] taken around the measure borders, that is the actual transition regions, many cuts now

[4] Using the Hanning Window.

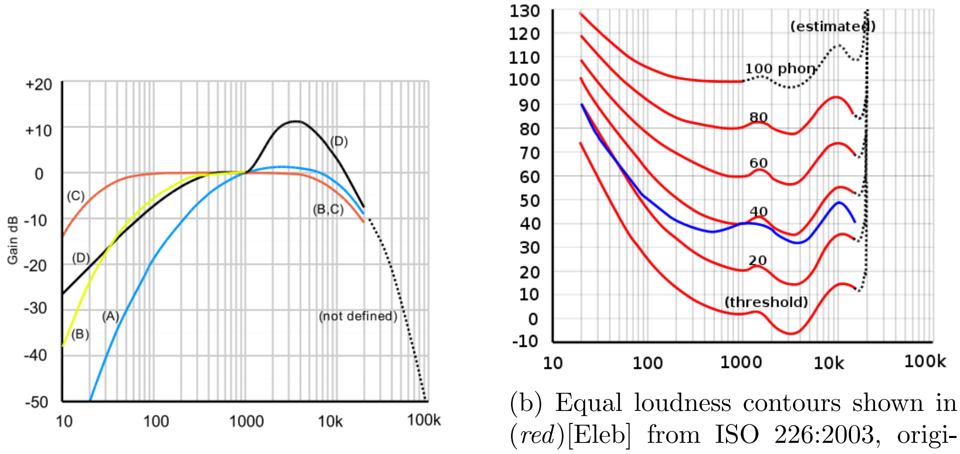


(a) Visualization of cuts.

(b) Visualization of path.

(c) Self-similarity matrix showing pair-wise similarity between analysis windows.

Figure 4.3: Results of Refined Comparison Metrics (same example song as above)



(a) Acoustic weighting curves [Elea]: A-weighting (*blue*), B (*yellow*), C (*red*), D (*black*)

(b) Equal loudness contours shown in (*red*)[Eleb] from ISO 226:2003, original ISO standard shown in *blue* for 40-phons, resembling sound pressure level over frequency spectrum

Figure 4.4: Weighting curves and equal loudness contours.

match perceptually smoother, as not only the measure contents are compared to each each other. As a side effect the problem of simple repetitions of a song is solved by ignoring the first and last measure during analysis.

4.3 Respecting Perception

For the further enhancement of the cut quality, we take the human sound perception characteristics into account during the similarity analysis phase. The ability of hearing is not solely a mechanical phenomenon of alternating pressure travelling through the air but rather a sensory and perceptual event. Mechanical sound waves arriving at the listener's ear are transformed to neural action potentials for further processing within the brain.

The scientific field of *psychoacoustics* models the transformation of physical signals towards an auditory impression in sequential steps, linked to the human's ear and cognitive signal processing [Ste03]. The research therefore focuses on the relation between objectively physical stimuli and their human perception to model hypotheses on auditive processing. It is advantageous not to only consider the mechanics of an environment but also the connection of these to the human ear and brain, involved with the listening experience. The ear has a non-linear response regarding different sound intensity levels called loudness. Loudness defines the attribution of auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud. Different filters have been proposed to adjust measured sound events to perceived loudness of the average human.

The *A-Weighting* curve is one of four curves *(A, B, C, D)* for loudness perception adjustment (Figure 4.4a). These different curves account for different loudness of sound in ascending order, from A used for normal environments up to D for loud aircraft noise. The A-Weighting curve derives from the 40-phon equal loudness contour proposed by *Fletcher and Munson* in 1933 as an approximation of its inversion to resemble gain (Figure 4.4b). The original investigation explored the relationship between loudness levels and the ear's varying frequency response. Volunteers were exposed to pure tones of varying frequency and were asked to adjust the sound levels to a reference tone. According to the results, the human ear is less sensitive to low frequency sounds. This experiment has been repeated in 1956 by *Robinson and Dadson* resulting in different curves that were considered to resemble the response of the human ear more accurately and led to the ISO226 Standard [ISO03]. Doubts on their validity raised the demand for another survey and resulted in a new revision of the standard as of 2003, showing the close similarity of current research and the 40-phon contour by Fletcher and Munson that provided the base for the A-Weighting filter:

$$A(f) = \frac{12200^2 \cdot f^4}{(f^2 + 20.6^2) \quad \sqrt{(f^2 + 107.7^2)\,(f^2 + 737.9^2)} \quad (f^2 + 12200^2)} \tag{4.2}$$

Loudness perception generally is a much more complex task than just A-Weighting [Ols72], but it delivers a good approximation sufficient for our application. We therefore apply the A-Weighting to our analysis window with the same setup as in the previous section. After computing the Fourier Transformation of the window, the energies of the corresponding frequencies are obtained. Now the sound pressure can be calculated as the local deviation from the ambient atmospheric pressure.

$$p = fA$$

The logarithmic sound pressure level measure in decibel is given by

$$L_p[dB] = 20 \cdot log_{10} \frac{p}{p_0}$$

The A-Weighting curve is now applied to the sound pressure level

$$L[dB(A)] = L_p[dB] + A(f)$$

and then transformed back into sound pressure with

$$\tilde{p} = 10^{\frac{L[dBA]}{20}} p_0 = 10^{\frac{A(f)}{20}} \cdot p_0 \cdot 10^{\frac{L_p[dB]}{20}} = 10^{\frac{A(f)}{20}} p = 10^{\frac{A(f)}{20}} fA$$

We now compare the perceptually weighted analysis windows pairwise as described earlier to compute the self-similarity matrix of the input song.

4.3.1 Results by Perception

Retaining the positions of the highest similarities again improves the cuts quality. While the previous approach of placing the analysis window to overlap two measures resulted in significant improvements regarding harmonically correct cuts, A-weighting removed some of the incorrect cuts regarding instrumental depth or voice articulation.

With the perceptually weighted analysis, we are able to generate a novel soundtrack according to a user's constraints containing transitions that go mostly unnoticed by the listener.

Still some problems exist, for the most part only relevant to the trained listener, who might perceive, in case of being familiar with a song, some structural cut misalignments, for example the introduction of a percussive background pad or a similar repetitive but quietly embedded texture, that fits musically but is otherwise not present in the song's genuine succession. Differing voice articulations also account for badly executed transitions recognized by the professional listener, as the vocalist might raise their voice in the progression of the source measure and start out with the same at the destination.

The few cuts not matching completely do not break our approach for the resynthesis application, nor limit the cut search space too strictly, as it would be done by only allowing a very small set of cuts, matching every underlying instrumentation and articulation in every possible detail.

4.3.2 Roadmap

The results obtained until now enable the next step of constrained audio resynthesis: the reassembly of the segmented musical content to generate a novel soundtrack according to specific guidelines.

The next chapter presents approaches handling the navigation and rearrangement of the segments.

Chapter 5

Pathfinders

5.1 Navigating the Segmented Content

In the previous chapter (4) we described the retrieval of cuts, that are positions in a song, to allow transitions between each other due to their similarity going mostly unnoticed by the listener. A cut has a start and end position, defining the source and target of a jump within a song. The progression of the song segments selected according to user-defined rules is called a path. This path begins at the user-specified start position in the input song and runs along the start position of the first cut. The path is now continued from the end position of that cut until the start position of the next cut is reached and so on to finally end at the user-supplied path's destination position.

This chapter explains different approaches of stitching together the song segments, using methods running almost unattended to those requiring a certain level of human interaction. Most of these derive from the use-cases given in the introductory chapter (1) and are implemented in concept studies for evaluation as laid out in the next chapter.

5.2 Time Scaling

We start out with the most simplistic resynthesis application of scaling a musical piece in time by lining up matching segments to meet the constraints of a given song length.

5.2.1 Genetic Path

As we build on the results by [WM11], we have a set of algorithms for finding a path through the segments of a song at our disposal. The most mature and applicable algorithm of their collection is the *Genetic Path Algorithm* that has been described in-depth in [WM12]. It basically generates a huge number of path candidates, iteratively alters them, and selects the best one according to the specified length constraints. The algorithm quantifies the quality of a path by measuring an energy functional consisting of the comparison of the user-specified duration to the path length, the actual cut quality and segment repetition suppression term. The optimal path is found by minimizing this energy while generating a large number of paths with crossover and mutation iteratively selecting the best ones. The *Genetic Path Algorithm* is fed a number of the best cuts found and run with a user-defined output length constraint, like generating a soundtrack three times the duration of its input song. Due to the high quality cuts emitted by our methods described in chapter 4, the algorithm performs reasonably well on generating syntactically correct soundtracks.

The downside of this approach is its runtime speed. Genetic algorithms, due to their general problem solving strategy, are applicable to a large number of areas with a vast search space, but without necessarily knowing much about their specific case. The algorithm's performance is not sufficient for generating a novel soundtrack in time when it comes to real-time applications. We therefore demand a faster method for finding the path through a song.

5.3 Respect the Measure

The next step is to find an algorithm to target the computational speed issues encountered above and simultaneously respect user-supplied structural constraints. These constraints represent a rearrangement of manual song annotations on the measure level to describe how the resulting generated soundtrack's structure should look.

5.3.1 Structural Constraints with Genetic Path

Having been around and tested for quite a while, our first objective is to modify the *Genetic Path Algorithm* to pay attention to user-defined structural constraints. To accomplish this, we extended the energy functional by [WM12]

$$E_{\mathrm{w}} = E_{\mathrm{cut}} + E_{\mathrm{duration}} + E_{\mathrm{repetition}} \quad , \tag{5.1}$$

with a weighting term for comparing the user-annotated song structure ground truth to the user-given novel song structure defined by rearranging elements of the ground truth. Our structure penalty term segment-wise compares the annotations of the path generated by the *Genetic Algorithm* to the ground truth annotations and reads as follows:

$$E_{\mathrm{structure}} = \sum_{S \in \mathrm{path}} S_{gi} - S_i \tag{5.2}$$

That leads to the final energy functional

$$E_{\text{genetic}} = E_{\text{w}} + E_{\text{structure}} \quad . \tag{5.3}$$

Now it is possible for the user to give an input labelling for the original soundtrack and define a desired output label succession. The algorithm will attempt to find a path through the audio material with respect to the previously computed cut points to represent the user's desired output most closely.

Genetic Measure Results

When there are enough cut points available in the regions to reassemble, the algorithm will produce a novel song that only roughly matches the structure given by the user. The availability of cuts is the most profound weakness of this approach. Often severe problems occur due to the design of the cut detection algorithm. The *Genetic Path Algorithm* is ultimately restricted to a certain number of highest rated cuts by the cuts detection algorithm. The user cannot override cut decisions by stating his superior knowledge of certain part progressions. In most cases the user asks the algorithm to produce structural successions of different, and sometimes stretched, logical song parts. These best cuts allow closely matched simple output length constraints as described above but do not cover all portions of the song permitting jumps within every part. Allowing more cuts just leads to more cuts closely between the best ones (closing the lines as seen in 4.3a, visually spoken) and does not allow for more jump possibilities.

Due to these facts, the genetic approach generates somewhat inaccurate results when a novel song of a certain structure is requested. We therefore raise the demand for more precise results.

The speed issues remain to be addressed and the performance is far away from real-time.

5.3.2 Structural Constraints with Belief Propagation

We try to address the inaccuracies caused by the former genetic approach with *Belief Propagation*, also known as sum or max-product message passing. Belief Propagation, first proposed by [Pea82], is a message passing algorithm performing inference on graphical models like *Markov Random Fields (MRF)* or *Bayesian Networks* which have been successfully employed in *artificial intelligence* and *information theory*. We are relying on the max-product implementation by [TF03]; more information on Belief Propagation can be found in [YFW03].

$$E_{\text{datacost}} = \begin{array}{cc} & \begin{array}{cccccccccccc} A & A & A & A & B & B & B & B & C & C & C & C \end{array} \\ \begin{array}{c} A \\ A \\ B \\ B \\ C \\ C \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 5.1: Energy matrix representing the user-annotated structure of the input song (columns) and output song (rows). To fix a measure at a certian position, all other measures are to be penalized in that row.

To employ Belief Propagation for our scenario, we translate our problem to a MRF formulation that is defined by two energy terms:

$$\min_{i,j} \sum_{i,j} c(T_i, T_j) + \sum_{i} (1 - \delta(l_{T_i}, \hat{l}_i)) \quad . \tag{5.4}$$

The first term accounts for costs of input measure j after input measure i and is the self-similarity matrix computed in chapter 4. The second one describes the cost for input measure j at output position i as $n \times m$ binary matrix where n is the number of input measures and m the number of output measures.

This matrix also represents the user-annotated input and output song structure with the n-th column being the information whether the n-th input measure is allowed for the m-th output measure row (Figure 5.1).

Elements of ones penalize the usage of that specific measure while elements of zeroes allow it. Begin and end measures of a logical song part need to be fixed to ensure a smooth transition between them. Measures of non-adjacent but otherwise logical equal parts from the input may be allowed for an output measure.

Belief Propagated Results

The Belief Propagation approach delivers ultra-fast results within a second on a current home computer and is suitable for real time resynthesis, compared to the genetic algorithm that has a minimum runtime of about one minute. This approach directly operates on the measure-level and produces results on behalf of the user's request, with a synthesized target song containing exactly as many measures as defined.

In the case of the target song not simply being a rearrangement of the original musical large scale structure, that is shifting around the chorus and verses, a jump is forced with respect to the locally best cut points within a certain song part. When the user requests a defined part like a verse

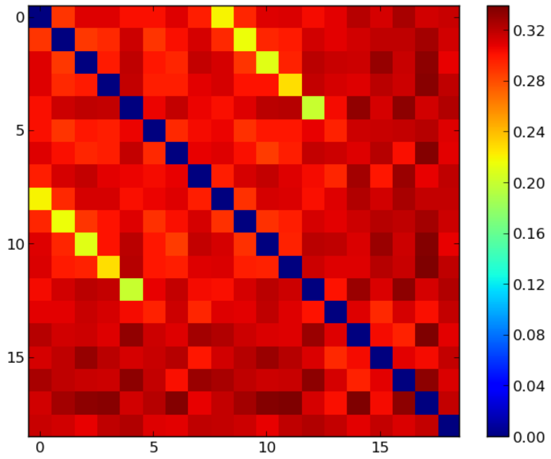


Figure 5.2: Self-similarity matrix of an isolated song part. Each of the ten best cut positions, forming the symmetrical secondary diagonals, only allow advancement or retrogression of exactly eight measures. Every extension for this specific part requires at least the addition of a multiple of eight measures to maintain smooth transitions.

or chorus to be extended by very few measures to fit their application, a transition may be noticeable by the listener due to the lack of highest quality cuts to allow the insertion of such a short segment. In most cases the local maxima within a song part lie a number of measures apart, so the user may want to take into account the minimum extension length a part requires to be seamlessly enlarged (Figure 5.2). This trade-off between desired measure number, that is song length, and highest transition accuracy cannot be eliminated by our approach as it would require further manual sound-technical processing and modification of the input song.

5.3.3 Time Scaling with Belief Propagation

Time scaling as done with the *Genetic Path* approach can also be done employing *Belief Propagation*. Similar to Figure 5.1, the user has to supply a matrix that describes the desired length constraint by fixing the first and last measure of the input song and filling the rest with no penalizing entries. Since Belief Propagation produces a song of exactly as many output measures as defined, the user has to select the length of erosion or dilation according to the distances of the best jumps as described in the previous section. Sticking to these guides, this approach produces good results, otherwise the selection of the best cuts will be overridden and cuts of diminished quality will be chosen leading to degradation of the overall quality of the synthesized target song.

Chapter 6

Results

This chapter gives an overview of the results that have been achieved using the methods described throughout this paper. So far, methods on finding and selecting good cut points in a song, as well as approaches for using these for reassembly of a novel song, have been proposed.

At first we summarize the quality of the acquired cut points in a song.

6.1 Quantifying the Cuts' Quality

We start out with an estimation of the quality of the cuts that have been found in Chapter 4. Due to the nature of music, quantification has to be done by a human listener. Audio snippets around the cut positions have been extracted and saved for manual analysis. A listener then rated the cuts according to three different levels. Cuts that expose strong internal rhythmic or harmonic mismatches noticeable by every untrained listener are rated with level one. Mismatches that account only for slight distraction and are recognizable only to the experienced listener are rated level two. Level three rating is assigned to cuts not recognizable by rhythmic, harmonic or melodic violations. For a comparison of the three different approaches of Chapter 4 we compare the 40 best cuts generated with each of the methods of each randomly selected song.

Table 6.1 shows the cut quality evaluated by a trained listener for our three incrementally improved approaches. Some songs' cuts improve heavily while switching from the first to second method using a measure-border analysis window whereas others are improved with perceptual weighting. The overall quality improves with every method advancement except for one example that yields slight harmonic misalignment in one cut with perceptual weighting.

Cuts Quality									
	Measure-wise			Transitions with Hanning Window			Transitions with Perceptual Weighting		
	+	o	−	+	o	−	+	o	−
Covenant – Lightbringer	31	3	6	40	0	0	40	0	0
Friska Viljor – In the Nude	19	5	16	34	5	1	36	4	0
Italo Brothers – Radio Hardcore	33	1	6	40	0	0	39	1	0
Neil Young – Southern Man	38	2	0	38	2	0	39	1	0
Owl City – Deer in the Headlights	34	2	4	40	0	0	40	0	0
Sunrise Ave – Hollywood Hills	30	9	1	30	9	1	36	4	0

Table 6.1: Resulting cuts' quality ordered by proposed methods

6.1.1 Comparison with Previous Results

This section gives a comparison of the results by [WM11] with results generated using our most mature perceptual approach and the same inputs. [WM11] used some input songs from a variety of genres, that are of a more pathological nature in our setting, as they are not directly related to dance, like classical performances, and therefore expose problems in beat pre-processing as described in Chapter 3.

Table 6.2 has the comparison in numbers. Even the classical example exposes a number of good cuts with our methods. The fundamental problem in this case is the pre-processing stage of beat tracking. In some song areas, the tracking works quite well as opposed to others with high dynamics or blurred note onsets. Incorrect measure beginnings often lead to consistent cuts when this incorrectly shifted phase has the same offset at the cut's target position. The same applies to the folk example. Difficulties arise due to timing variations and the musicians free interpretation and articulation of certain passages where the approach unaware of the musical structure has advantages. The Electronic, Hiphop and Punkrock examples give good results by their straight and homogeneous designs, especially the electronic one.

6.1.2 Cuts Quality Estimation of Contemporary Dance Music

The next estimation focuses on music that represents common styles often played in dance academies and rehearsal situations.

Comparison to Previous Results						
	Hierarchical Analysis [WM11]			Transitions with Perceptual Weighting		
	+	o	−	+	o	−
Bob Dylan – Blowing in the Wind (Folk)	27	7	6	22	7	11
Dendemann – Endlich Nichtschwimmer (Hiphop)	30	3	7	40	0	0
Digitalism – Zdarlight (Electronic)	19	9	12	39	1	0
Pyotr Ilyich Tchaikovsky – Valse in A (Classic)	38	0	2	29	2	9
Zebrahead – Playmate of the Year (Punkrock)	34	2	4	40	0	0

Table 6.2: Comparison of the songs used for evaluation by [WM11]

This song selection (Table 6.3) features a large number of dance styles included in the official training programme by the ADTV[1].

6.1.3 A Note on Computational Speed

All our evaluated approaches feature the *Fast Fourier Transform (FFT)* for distance comparison between analysis windows. This transform seems to be the bottleneck in our cut detection implementations as the runtime for different input songs ranges from 10 seconds to several minutes. Internal optimizations by the employed FFT implementation[2] may slow down the computation process due to size of characteristics of the analysis window. The performance achieved so far is not real-time but this step has only to be done once for every song. After loading the pre-computed results, a user can rearrange the input song according to their demand in real-time.

6.2 Reassembly results

We now summarize some thoughts on our path search approaches proposed in Chapter 5.

6.2.1 Genetic Path

The *Genetic Path* method has originally been proposed by [WM11] for this domain and has only been used and modified by us. This method proved to be inaccurate for our application to a certain extent, as it is restricted to a

[1] Allgemeiner Deutscher Tanzlehrerverband e. V.

[2] Numpy.fft

Contemporary Dance Music			
	Transitions with Perceptual Weighting		
	+	o	−
Slow Waltz	40	0	0
Slow Waltz	33	7	0
Viennese Waltz	35	5	0
Viennese Waltz	39	1	0
Quick Step	40	0	0
Samba	36	4	0
Cha cha	37	3	0
Disco Fox	40	0	0
Disco Fox	40	0	0
Foxtrot	38	2	0
Rumba	38	2	0
Jive	38	0	2

Table 6.3: Cut Quality of Contemporary Real-World Dance Music

given number of the best cuts, that do not allow for fine-tuned time scaling due to the nature of a song structure and target lengths will be matched only approximately. Figure 6.1 shows an example path through a song that has been stretched to circa 8 minutes.

We extended the algorithm's energy functional by a structural weighting term without great success. User defined target song structures were matched only roughly, rendering this approach, despite its runtime, unusable for dance floor applications. Rearrangement computation according to some users' specified structure rules should not take between a half and several minutes.

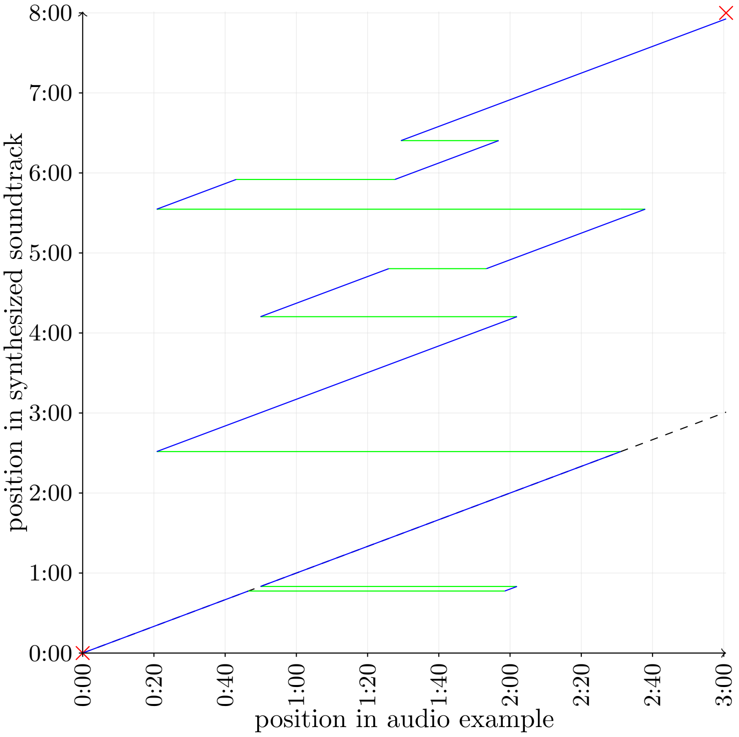


Figure 6.1: Visualization of path through *Deer in the Headlights* by *Adam Young* with time scaling to 8 minutes

6.2.2 Belief Propagation

With *Belief Propagation* we sped up the path computation while improving target song assembly with respect to structural output constraints. The user has to supply a measure-wise annotation of the input song structure and the desired target arrangement to get a path through the song in less than a second. This algorithm can be forced to produce a novel song of a certain number of measures by supplying it with the whole Self-Similarity Matrix. Modifications that music theoretically do not fit for the target structure can be made but will decrease the quality of the target, as low quality cuts have to be chosen.

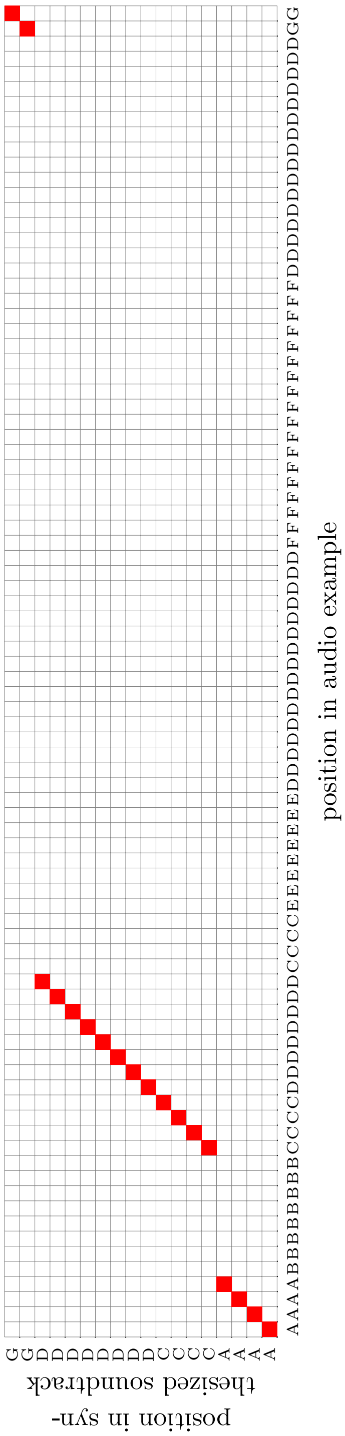


Figure 6.2: Visualization of path through *Vayamos Companeros* by *Marquess*. Red boxes show positions of input measures in the synthesized target song

Chapter 7

Conclusion and Discussion

This chapter finalizes this thesis by taking a look at achievements, pitfalls, milestones and the way, our work has developed. We successfully proposed a novel audio resynthesis approach, based on the rhythmical structure of an audio source.

The first part comprised the detection of cut points, that is the start and end position of a jump in the musical source, while the second part extended existing methods of song rearrangement and presented a novel reassembly approach.

7.1 Thoughts on Measure-Based Audioresynthesis

Three incrementally improved methods in the field of similarity analysis for audio resynthesis have been proposed. Beat tracking gave a great reduction of search space, as now only defined transitions—the measure boundaries—are to be considered as jump positions in the musical source.

At first, we compared the whole measure contents pair-wise, found by the underlying beat tracking technology[1], with good initial results (see Chapter 6), proving the structural approach appropriate for this application.

In the second step the analysis window has been shifted from the measure contents to the measure boundaries for better transitions, and weighted with the Hanning function to smooth out the signal at its borders. Boosting the positive results, still some outliers remain to be addressed.

The third step perceptually weighted the window as described in the previous iteration by applying *A-Weighting*, a human loudness perception curve according to [ISO03]. This last iteration removed many remaining cuts that could cause harmonic or transitional irritation to the listener.

Whilst we produced many very good results, the cut points search is only as good as its preprocessing step of tracking the beat. With still some, for the trained listener, slightly notable jumps left, we achieve an overall

[1] [aufTAKT], www.zplane.de

performance on a real-world application enabling level for modern music, especially dance music.

7.2 Thoughts on Song Reassembly

The follow-up of the cut points search is the task of finding a good path through the now segmented musical content for which we presented two approaches throughout this thesis. Our first and obvious approach was to use the existing *Genetic Path Algorithm* [WM12] for the rather simple use case of extending or shrinking the length of the input song to a defined length. While the genetic algorithm produced good reassembly results due to the selection of the best high quality cuts produced in the previous steps, it is exclusively limited to the number of supplied cuts and therefore might not be able to fully meet the user-defined length constraint. In case of bad cuts, that are cuts that will be noticed even by the untrained listener, the algorithm will also produce results according to the previous step's performance. Its run-time speed is somewhat high compared to the user's patience waiting for results, due to its general problem solving capabilities that lack specific knowledge on the certain task to be solved.

The second, more complex use case involved not only scaling the input song in time, but also rearranging it to produce an output song following certain structural succession constraints. To accomplish this goal, the energy terms of the genetic path approach have been extended by structural measure awareness to enable the user to supply measure-wise input and output structure annotations. This attempt turned out not to work well, as the still good sounding target song did not—or only roughly—resemble the user's desired target annotation.

These limitations are caused mainly by the absence of cuts within the desired transition regions, that is directly between the actual song parts. A further problem is the conflict of objectives between the structural energy and the repetition energy origination from the genuine energy description by [WM12], but only in a theoretical sense, as disabling it does not improve results significantly.

Enter *Belief Propagation.* To overcome the limitations experienced using the genetic path algorithm, we proposed the usage of the Belief Propagation Algorithm originally introduced by [Pea82], that proved appropriate to solve the path search problem in real-time with measure-accurate results. Measure-wise input and output annotations have to be supplied that may force jumps at positions where the user knows them to work correctly for their purpose. Annotated song parts may be removed, scaled or shifted around, with the musical design limitation that unnoticeable cuts within a part may lie a certain number of measures apart, so a part may only be seamlessly scaled with respect to that factor.

Informing the user about these scaling factors according to their annotated song structure leads to the synthesis of novel soundtracks without auditory violations when following these hints.

7.3 The Big Picture

A user, like a dance choreographer as intended by our main use cases, is now given a tool to aid the construction of a novel soundtrack according to their structural demands. Without deeper knowledge about sound editing practices, large structural changes can be made to a song, solely with knowledge about the song's basic architecture, that is the idea of what is beat, what is a measure and what is the difference between chorus and verse.

7.4 Road Map

Software tends to evolve over time, so advances in the field of beat tracking and recognition may broaden the usage of this approach to a wider variety of musical genres as well as stabilize results on existing genres known to work rather well.

The simplification of the musical input annotation process by structural recognition [PMK10] may aid the user to segment a song faster. Music structure analysis still has severe flaws, but even today may give an initial hint about a song's part distribution.

These ideas together with the results presented throughout this thesis integrate into a fully functional application ready to be deployed in the non-academic world to fit the use cases described earlier.

Improved automatic song annotation, solutions for labelling parts of a song according to their perceived quality and features like fast, slow, energetic, loud, ambient etc., may come up over time, and enable building huge databases of music that can be retrieved to reassemble a novel soundtrack with respect to some descriptions of high level features.

Solutions for automatic song generation to accompany a silent video may come to light, that take into account the mood perceived in the video, and then query a database as described before, for some matching music segments to be rearranged to generate a novel soundtrack.

The road to future developments converges as a small, distant point towards the horizon, alongside which other great ideas will arise within this research area.

Chapter 8

Acknowledgements

As one of the most important parts of this work, this chapter should not be overlooked. It has been a long journey to make this thesis a success and there have been many helping hands to support me whenever I got stuck.

My very first, profound expression of gratitude goes to Stephan Wenger, my supervisor. Without his technical and personal support, his expertise, devotion and duty, my work would not have come to light in this way. He was the one who was always available for questions and requests, ready with open ears to listen to every idea and thought, helping me through all my deep mental caves.

I also would like to express my deep commitment to *zplane.development GmbH & Co. KG*, especially to zplane's Tim Flohrer and Alexander Lerch, for kindly providing *[aufTAKT]* the core technology that made my research possible. zplane is a research-focused company that provides state-of-the-art music processing / analysis technology and know-how for the music industry[1].

Special thanks to my friend Holiday Bannister, my language supervisor, for her support from England. She did all the linguistical corrections of my work from a native speaker's point of view. Her work has been incredibly helpful during the creation of this thesis.

I got much support from a department very distant to traditional science, the team of the dance academy *Haeusler Kwiatkowski* who provided the non-academic input necessary for this thesis. The head-choreographer *KO* Kirsten Hupe shared her views on dance theory and software requirements and suggested scenarios for field tests. Thank you to Chriss Melzer for expanding my views on the area of dancing academy communities and their demands, and to Jan-Peter Heemsoth for in-field testing during evening dance events. Their collaboration and assistance greatly improved the quality of my results.

Every modern thesis needs to be written on a computer. Due to the

[1] http://www.zplane.de/products/company/profile

shortcomings of a mobile version, a big thanks goes to Christoph Weckmann for providing me with one.

For a broader view, some analysis is better done by someone not directly related to one of the covered subjects. I would like to thank Sebastian Bode for doing in-field requirements analysis of dance academie's demands.

Thank you to Anja Franzmann, the *CG Lab*'s bureaucratic manager, for helping me with seamless in-university human interaction. She kept away from me most of the confusing and time-consuming red tape.

Also I would like to thank Prof. Dr. Marcus A. Magnor, head of the CG Lab, for giving this work an academic platform.

I am grateful to Thomas Neumann, a fellow researcher at CG Lab, for his algorithms and graph visualization support.

Finally, thank you so much to all of the other residents of CG Lab, for making the work at the Institute a great experience and pleasure.

Bibliography

[BC57] J.R. Bray and J.T. Curtis. An ordination of the upland forest communities of Southern Wisconsin. *Ecological monographs*, 27(4):325–349, 1957.

[BDA$^+$05] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[BTTT59] R.B. Blackman, J.W. Tukey, J.W. Tukey, and J.W. Tukey. *The measurement of power spectra: from the point of view of communications engineering*, volume 1058. Dover Publications New York, 1959.

[CBR03] M. Cardle, S. Brooks, and P. Robinson. Audio and user directed sound synthesis. In *Proceedings of the International Computer Music Conference (ICMC), Singapore*, 2003.

[Cha05] W. Chai. *Automated analysis of musical structure*. PhD thesis, Massachusetts Institute of Technology, 2005.

[Col08] K. Collins. *Game sound: an introduction to the history, theory, and practice of video game music and sound design*. The MIT Press, 2008.

[CT65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[Dix01] S. Dixon. An empirical comparison of tempo trackers. In *Proceedings of the 8th Brazilian Symposium on Computer Music*, pages 832–840, 2001.

[Dix07] Simon Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–50, 2007.

[DP07] M.E.P. Davies and M.D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, 2007.

[Ear07] A. Earis. An algorithm to extract expressive timing and dynamics from piano recordings. *Musicae Scientiae*, 11(2):155, 2007.

[EKR87] J.P. Eckmann, S.O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters (EPL)*, 4:973, 1987.

[Elea] Lindos Electronics. A-weighting in detail. *Lindos Website.*

[Eleb] Lindos Electronics. Equal-loudness contours. *Lindos Website.*

[Fer] Brian Ferneyhough. The Ensemble Sospeso. Web Archive: `http://web.archive.org/web/20110721014850/http://www.sospeso.com/contents/articles/ferneyhough_p1.html`.

[Flo11] T. Flohrer. [auftakt] SDK 3.0.2 documentation. 2011.

[Foo99] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80. ACM, 1999.

[GD05] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[GM94] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the second ACM international conference on Multimedia*, pages 365–372. ACM, 1994.

[GMS10] P. Grosche, M. Müller, and C.S. Sapp. What makes beat tracking difficult? A case study on Chopin mazurkas. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), Utrecht, Netherlands*, pages 649–654, 2010.

[ISO03] Acoustics normal equal loudness-level contours. British Standard ISO 226:2003, International Organization for Standardization, 2003.

[KEA06] A.P. Klapuri, A.J. Eronen, and J.T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.

[LHL10] S. Lefebvre, S. Hornus, and A. Lasram. By-example synthesis of architectural textures. *ACM Transactions on Graphics (TOG)*, 29(4):84, 2010.

[LLW$^+$02] Lie Lu, Stan Li, Liu Wenyin, Hong-Jiang Zhang, and Yi Mao. Audio textures. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–1761–II–1764, 2002.

[LWZ04] L. Lu, L. Wenyin, and H.J. Zhang. Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, 12(2):156–167, 2004.

[McC05] J. McCormack. Open problems in evolutionary music and art. *Applications of Evolutionary Computing*, pages 428–436, 2005.

[MMDK07] MF McKinney, D. Moelants, MEP Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[Ols72] Harry F. Olson. The measurement of loudness. *Audio Magazine*, pages 18–22, 1972.

[PB04] JR Parker and B. Behm. Creating audio textures by example: tiling and stitching. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, volume 4, pages 317–320. IEEE, 2004.

[Pea82] J. Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach.* Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.

[Pee07] G. Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Applied Signal Processing*, 2007(1):158–158, 2007.

[PMK10] J. Paulus, M. Müller, and A. Klapuri. State of the art report: Audio-based music structure analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 625–36, 2010.

[Ros92] David Felix Rosenthal. *Machine rhythm: Computer emulation of human rhythm perception.* PhD thesis, Massachusetts Institute of Technology, Dept. of Architecture, Cambridge, MA, USA, 1992.

[RP07] A. Robertson and M. Plumbley. B-keeper: A beat-tracker for live performance. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 234–237. ACM, 2007.

[S^{+}00] D. Schwarz et al. A system for data-driven concatenative sound synthesis. In *Digital Audio Effects (DAFx)*, pages 97–102, 2000.

[SA10] J.M. Smith-Autard. *Dance composition: A practical guide to creative success in dance making.* Methuen Drama, 2010.

[Sch05] D. Schwarz. Current research in concatenative sound synthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 9–12, 2005.

[Sch06] D. Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3–22, 2006.

[SDP09] A.M. Stark, M.E.P. Davies, and M.D. Plumbley. Real-time beat-synchronous analysis of musical audio. In *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy*, pages 299–304, 2009.

[SERlG06] G. Strobl, G. Eckel, D. Rocchesso, and S. le Grazie. Sound texture modeling: A survey. In *Proceedings of the 2006 Sound and Music Computing (SMC) International Conference*, pages 61–5, 2006.

[Ste03] Jonathan Sterne. *The audible past: Cultural origins of sound reproduction.* Duke University Press Books, 2003.

[Stu04] B.L. Sturm. Matconcat: an application for exploring concatenative sound synthesis using MATLAB. *Proceedings of DAFx04, Naples, Italy*, 2004.

[TF03] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 900–906. Ieee, 2003.

[TP80] J. Tenney and L. Polansky. Temporal gestalt perception in music. *Journal of Music Theory*, 24(2):205–241, 1980.

[WM11] Stephan Wenger and Marcus Magnor. Constrained example-based audio synthesis. In *Proc. IEEE International Conference on Multimedia and Expo (ICME) 2011*, July 2011.

[WM12] Stephan Wenger and Marcus Magnor. A genetic algorithm for audio retargeting. In *ACM Multimedia*, 2012. To appear.

[YFW03] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

[ZP01] A. Zils and F. Pachet. Musical mosaicing. In *Digital Audio Effects (DAFx)*, 2001.