Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

Audio Resynthesis on the Dancefloor: A Music Structural Approach

Jan-Philipp Tauscher, Stephan Wenger and Marcus Magnor

Institut für Computergraphik, TU Braunschweig, Deutschland

Abstract

We propose a method for synthesizing a novel soundtrack from an existing musical piece while preserving its structure and continuity from a music theoretical point of view. Existing approaches analyze 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 but fail to maintain musically correct song progression. Introducing the alignment of rhythmic and harmonic structures during transition point detection, we employ beat tracking as the analysis core component and take the human sound perception into account. Automatic segment rearrangement is improved by employing a novel belief propagation approach that enables user-defined constraints for the output soundtrack, allowing video editors or dance choreographers to tailor a soundtrack to their specific demands.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications— H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing—Signal analysis, synthesis and processing J.5 [Arts and Humanities]: Performing Arts—

1. Introduction

Music as composed and recorded by an artist is of fixed duration and chronological construction, and is in some way carved in stone. At the same time, it plays an important role in the presentation of visual content like movies, but is often not directly usable as it has to be rearranged to fit its purpose. Utilizing the approach of example-based audio synthesis, inspired by retargeting of graphical textures, we propose a method for completely automatic or user-supported rearrangement of regularly structured musical compositions with respect to usability in motion picture scores, media support, electronic games and dance related institutions such as academies, theatres or clubs.

Often a specific song's runtime does not match its intended usage duration so that it becomes necessary to stretch or shorten a song by rearranging its parts. In the same context, one might also wish to create a medley song from parts of different songs, insert measure-aligned silence, or perform other restructuring operations, all without introducing any artifacts that become noticeable to the listener.

Our approach segments a soundtrack into its elementary logical parts, its measures, and uses self-similarity analysis to generate a list of possible seamless transitions between these parts. The parts are than automatically rearranged according to user-specified constraints to generate the final beat-aligned, structure preserving target soundtrack. In the simplest case, the user only needs to supply a target song length by defining an output measure count or a time length. For a more fine-tuned song-assembly, per-measure annotations allow for specific part selection during the synthesis phase.

The human cognitive processes that lead to our understanding of the underlying pattern of a musical performance cannot easily be reproduced by machinery, but current technology provides the basic tools necessary for machine aided understanding and restructing of musical pieces. In our work, we focus on music that can be analyzed and segmented with current technologies like beat trackers [Ros92] and structure analysis tools [Sch06]. The analysis and synthesis of sound consisting of textured patterns without defined pitch, rhythm, dynamics and timbral qualities (e.g., [LWZ04]) is beyond the scope of this paper.

2. Related Work

The contemporary *concatenative audio resynthesis* approach of cutting and stitching started in the 1950s and has been the

DOI: 10.2312/PE.VMV.VMV13.041-048

subject of active research until recently [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. However, they do not capture the musical structure of the source and therefore typically 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 patterns but not on music [CBR03]. *Audio textures* [LLW*02] segment input audio into short clips that are subsequently rearranged. Similar to concatenative audio resynthesis, these methods are typically not applicable to music. [SERlG06] summarize similar sound texture generation methods.

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.

In order to make audio resynthesis techniques applicable to music and other audio containing large-scale structure, [WM11] introduced a multi-resolution scheme for fast self-similarity analysis capturing the sonic source from multiple measures down to single samples, allowing perfect alignment of the cuts without blending or scaling. A genetic algorithm [WM12] was subsequently proposed to rearrange the segments into soundtracks subject to user-specified constraints. This approach performs reasonably well even on structured music, but often exhibits noticeable artifacts with respect to rhythmical structure: the proposed transitions do not always correspond to the underlying meter of the musical source, causing profound irritation to the listener or dancer who expects temporal continuity. The same applies to [WBSH*13], who analyse the music for its physical beat locations and super structures but do not always manage to obtain rhythmically correct transitions. Another problem are jumps within the track occuring at positions that fit harmonically but not at its musical depth level, like solo vs. tutti instrumentation, or do not match well at all. In addition, vocal parts are typically manually excluded from the resynthesis to avoid artifacts.

Audio-based music structure analysis [GM94, Cha05, PMK10] aims to segment the audio track into its logical pieces defined by the underlying beat, human perception and music theory. Using similar methods, we decompose a musical piece into its basic structural units, the measures, and automatically rearrange them into a novel soundtrack according to user constraints. In our approach, the user selects an input track and supplies constraints in the form of a cost function for each measure of the output. A beat tracker is used to segment the input audio, and our algorithm finds suitable transitions between the segments. Finally, it automatically generates an optimal sequence of measures that are finally reassembled to form the target soundtrack.

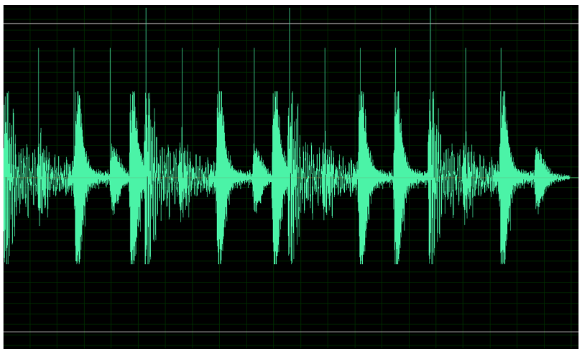
Figure 1: A sample drum loop with estimated onsets marked. Long lines show the down beats [Flo11].

3. Beat Tracking

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 (like $\frac{4}{4}$, $\frac{3}{4}$, $\frac{6}{8}$), usually contributed by percussion instruments or noticeable through chord changes or note alignment.

The recognition of beat in music in usually divided into certain phases. Beat tracking starts with the detection of note onsets in the song's signal, that is, computing a *novelty curve* for recording changes in energy, spectral content or pitch [BDA*05, DP07, Ear07, KEA06]. From these possible onsets, the beats are extracted by some kind of selection mechanism [BDA*05]. 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. Appropriate sequential beat positions for a correct description of the piece's periodic beat structure (Figure 1) are selected, regarding frequency of tempo and phase of timing.

A beat is a perceptual phenomenon and does not necessarily correspond to physical beat 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. [Dix01, MMDK07] give an overview of empirical evaluations of several beat tracking approaches.

Recognizing the beat in a given piece of music is crucial for our 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. To be usable in an audio resynthesis application, a beat tracker has to fulfill the following tasks:

Analyse and record beat structure: The ability to correctly follow the beat of a song, as a human 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. For popular music genres, it is typically sufficient to focus on simple time signatures.

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, leading to noticeable artifacts at the measure boundaries during resynthesis.

After evaluating the performance of various beat trackers, like *BeatRoot* [Dix07], *beatsync* [Che02], *BTrack* [SDP09], *B-Keeper* [RP07], and the tracking components available for the *Sonic Visualizer* [CLSB06] software, we selected the *[aufTAKT] tempo and beat tracking system* by *zplane.development* [Zpl13]. It was the only beat tracker able to reliably track the beat signal in our experiments and has 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, determines the time signature, finds the first down beat of a measure and adapts to varying tempo of a musical source.

[aufTAKT] has difficulties with material containing time signature changes, lacking regular beat patterns or featuring other experimental or uncommon musical properties. In most cases, this is irrelevant to us as we focus on trackable music for both motion pictures and dance that follows some assumptions like static time signatures, defined song building blocks and an approximately steady tempo. Some soundtracks used for movie scores may violate these assumptions, but in many cases, the same audio properties that make a piece difficult to track for a computer are also less obvious to a human listener, so that erroneous transition may still go unnoticed.

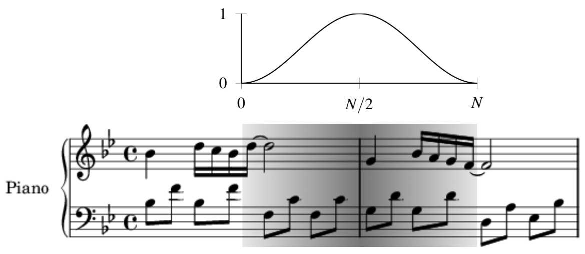

Figure 2: Analysis window and its position on a measure boundary.

4. Audio Analysis on the Measure Level

The beat tracker described in the previous section provides sample-accurate beat positions and measure boundaries of the input song. Based on this structural information, we perform a self-similarity analysis of measure transitions to find measure boundaries where a jump to an alternative successor can be introduced without noticeable artifacts. The general concept is to divide an audio source into into a useful sequence of features $(x_1, x_2, ..., x_n)$, compare its elements pairwise according to some distance function d, and store the results in a self-similarity matrix $C_{i,j} = d(x_i, x_j) : i, j \in 1, 2, 3, ..., n$. This concept was already employed by [EKR87] for analysis of chaotic systems and was later introduced in the domain of music analysis for visualization of an audio recording's time structure [Foo99]. We compute the self-similarity via the *Bray-Curtis* dissimilarity [BC57],

$$d(u, v) = \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|} \quad , \qquad (1)$$

as distance function. It is one of the most well-known non-metric ways of quantifying the difference between data sets and delivers robust and reliable dissimilarity results throughout many applications.

As the analysis window for computing measure-boundary similarity, we employ a Hanning window [BT59] centered at measure boundaries, Figure 2. The Hanning window, defined as

$$w(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \quad , \qquad (2)$$

smoothly transitions from one in the center to zero at the borders. This ensures that the actual jump region—the musical bar—is weighted most.

For the further enhancement of the cut quality, we take the human sound perception characteristics into account during the similarity analysis phase. The ear has a non-linear response regarding different sound intensity levels called loudness that defines the attribution of auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud. The *A-Weighting* curve is one of four loudness perception adjustment curves accounting for different loudness of sound in ascending order, from A used

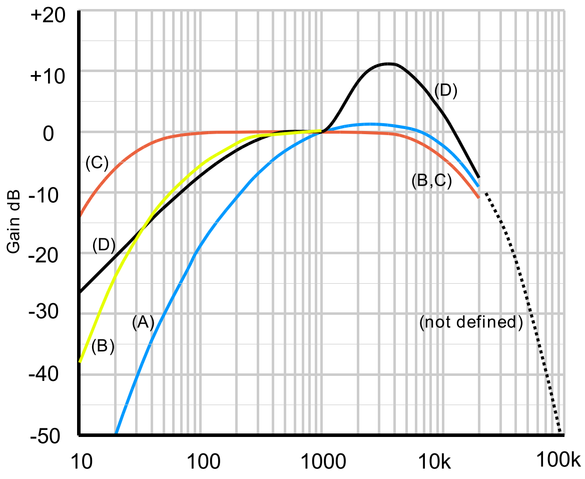

Figure 3: Acoustic weighting curves [Lin]: A-weighting (*blue*), B (*yellow*), C (*red*), D (*black*).

for normal environments up to D for loud aircraft noise, Figure 3. It is derived from the 40-phon equal loudness contour proposed by [FM33] as an approximation of its inversion to resemble gain.

$$A(f) = \frac{12200^2 \cdot f^4}{(f^2+20.6^2)\,(f^2+12200^2)} \cdot \frac{1}{\sqrt{(f^2+107.7^2)\,(f^2+737.9^2)}} \tag{3}$$

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. This accounts for good results when jumping within vocal parts or parts that differ in instrumentation but are otherwise harmonically similar. Then we compute the amplitude spectra of the length-equalized measures using a *Fast Fourier Transform* [CT65]. The logarithmic sound pressure level measured in decibels 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,

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

We now compute the Bray-Curtis distances between pairs of perceptually weighted analysis windows as described above to obtain a self-similarity matrix of the measure boundaries of the input song, Figure 4.

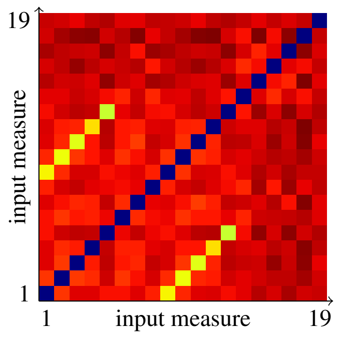

Figure 4: Self-similarity matrix C_{l_{i-1},l_i} of an isolated song part. Blue marks regions of high similarity, while regions of low similarity are indicated in red. Each of the ten best cut positions, forming the symmetrical secondary diagonals, only allow advancement or retrogression of exactly eight measures. Every extension for this specific part requires the addition of a multiple of eight measures to maintain smooth transitions.

5. Resynthesis

The self-similarity matrix provides a measure for the quality of transitions between any pair of measures in the input audio. In addition, the user provides a *constraint matrix* that for each output measure associates a cost for each measure from the input audio. This matrix can be of very simple structure, for example, when only time stretching of the audio is required (Figure 7); it can also be based on part annotations (Figure 5) or chosen more freely (Figure 8). Together, these measures describe the quality of any arbitrary sequence of input measures of the desired length. The goal of the automatic resynthesis step is to find a sequence of input measures that optimizes this quality.

Previous approaches like the *genetic path algorithm* introduced by [WM12] expose some flaws regarding runtime and applicability to our problem definition, as their search space was much larger due to their employed song segmentation strategy. Our measure-wise segmentation allows us to employ a *belief propagation* algorithm instead. Belief propagation [YFW03, TF03], 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. To employ belief propagation for our scenario, we write down the optimization problem as the sum of pairwise and separate energy terms,

$$\min_l \sum_i C_{l_{i-1},l_i} + \sum_i D_{i,l_i} \ . \tag{4}$$

$$D_{i,l_i} = \begin{array}{c|cccccccccccc} & A & A & A & A & B & B & B & B & C & C & C & C \\ \hline A & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ A & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ B & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ B & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ C & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ C & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}$$

Figure 5: Constraint matrix representing the user-annotated structure of the input song (columns, from left to right) and output song (rows, from top to bottom). To fix a measure at a certain position, all other measures are penalized in that row (binary example).

The first term accounts for costs of input measure l_i after input measure l_{i-1} and corresponds to the self-similarity matrix C computed in Section 4. The second term describes the cost for input measure l_i as output measure i and is expressed by the user as an $n \times m$ matrix D, where n is the number of input measures and m the number of output measures, Figure 5.

Non-zero elements of D penalize the usage of that specific measure while zero elements allow it. Begin and end measures of a logical song part can be fixed to ensure a smooth transition between them; typically, the first and last measure of the output will be fixed to the first and last measure of the input, respectively. Measures of non-adjacent but otherwise logical equal parts from the input may be allowed for an output measure.

The matrix D 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. Using this matrix, the user is able to specify the desired structure if the output song implying its length. In case of simple time stretching or shrinking, the user simply supplies 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. Using this approach, a novel soundtrack is generated according to the user's constraints.

6. Results

We start out with an estimation of the quality of the transitions that have been judged optimal by the self-similarity matrix computation. Due to the subjective nature of music, the quality has to be judged by a human listener. We assume that only the fourty best-rated possible transitions will be chosen by the belief propagation algorithm. Audio snippets of transitions across the fourty best-rated measure-boundary pairs were extracted and saved for manual analysis. An experienced musician then categorized the transitions into three different quality levels, Table 1. Transitions that show strong internal rhythmic or harmonic mismatches that would be noticeable even to an untrained listener are rated negatively. Mismatches that account only for slight distraction and are recognizable only to the experienced listener are rated neutral. A positive rating is assigned to transitions containing no noticable rhythmic, harmonic or melodic violations. Since one important use case of our method is the rearrangement of dance music for choreographies, our song selection features a large number of dance styles included in the official training programme by the ADTV (German Dance Federation).

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

Table 1: Transition quality of 40 best transitions found in several pieces of contemporary dance music. Cuts rated "+" were rated unnoticeable, those rated "○" could be noticed by an experienced musician, while cuts rated "−" contained severe artifacts.

Table 1 shows that for most songs, no strong artifacts were detected. Less severe artifacts were present in more than half of the songs. This category of artifacts would be acceptable, for example, in a dance class rehearsal setting. Where superior quality of the result is required, the corresponding entries in the self-similarity matrix can easily be manually penalized if noticeable transitions occur in the final result.

After the self-similarity matrix has been computed, the user supplies a measure-wise annotation of the input song structure and the desired target arrangement. The optimal output soundtrack can then be computed in less than a second. Our algorithm produces a novel song of the specified number of measures by means of the automatically generated self-similarity matrix. Modifications that violate the musical structure of the source can be created but decrease the quality of the target, as low quality cuts have to be chosen. In this case, 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 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

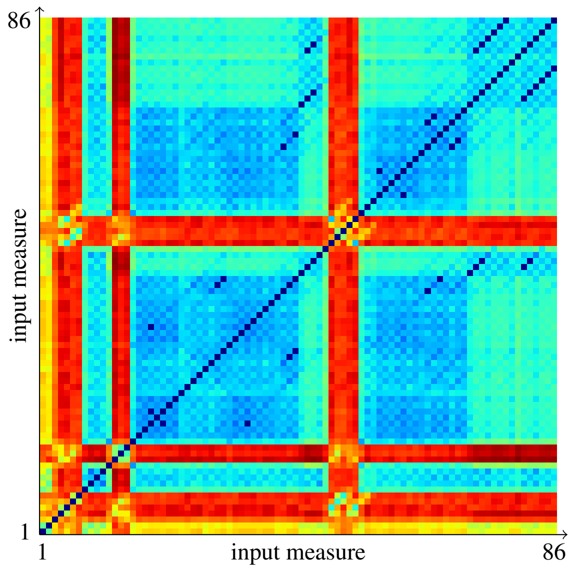

Figure 6: Self-similarity matrix of the song *Partyfreunde* by *Dave Sünti*. The forty best cut positions are shown as the symmetrical blue secondary diagonals and single blue spots. Their distances to the main-diagonal account for the scaling factors. Again, blue and red mark regions of high and low similarity, respectively.

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 4). 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.

The following example gives an overview of the resynthesis process. Figure 6 shows the self-similarity matrix of an input song. Since our previous evaluation has shown that typically, the forty best transitions are equally acceptable, they have been set to zero to avoid repeatedly choosing the best transition over equally good ones, which would result in unnecessary loops in the output. Its now the users' turn to supply an input annotations matrix, with columns representing the input song structure from left to right and rows representing for the output structure from bottom to top. In the simplest case, no input structure annotation has to be provided leading to an almost zero matrix that is framed by ones to fix the first and last measure, Figure 7. In our example, we provided a song annotation to fix the position of certain parts to create music for a simple discofox choreography, Figure 8.

Beginning at the start of the input song, we requested to

$$\begin{pmatrix} 0 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \ddots & & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 & 0 \end{pmatrix}$$

Figure 7: Constraint matrix representing only input and output song length for simple time scaling. The first and last output measures are fixed while the others are freely selected by the algorithm.

line up the intro and interlude part to be followed by some verse part, two times the length of one original verse, that may be taken from any of the input verse parts. After the verse part, we would like to have an instrumental bridge part lined up. The bridge part is then to be followed again by a verse in its original length but this time only material from the first and second one is permitted. As the last part we request a one-and-a-half sized variation of the chorus with outro parts, including material from the other part of the same variation that slightly differs in its sound atmosphere. Together with the transition cost information provided by the self-similarity matrix, the constraint matrix is then used by the belief propagation algorithm to compute an output succession of measures to assemble the target song (Figure 8, marked measures).

Audio files and additional resynthesis examples are presented in the supplementary material.

7. Conclusion, Discussion and Future Work

We proposed a novel audio resynthesis approach based on the rhythmical structure of an audio source. It comprises the detection of appropriate transitions between different positions in a song and a belief propagation resynthesis algorithm that rearranges parts of the song according to the transition qualities and additional user constraints. Our method employs beat tracking and perceptual weighting to respect the musical meter and the human perception. We use an advanced beat tracker to decompose the input audio not only into beats but into semantically meaningful measures. This not only ensures the rhythmical continuity of the output but also leads to a great reduction of search space, as now only defined transitions—the measure boundaries—have to be considered as jump positions in the musical source.

While our algorithm produces many very good results, the quality of the cut points search is limited by the beat tracking preprocessing step. With only a few jumps left that a trained

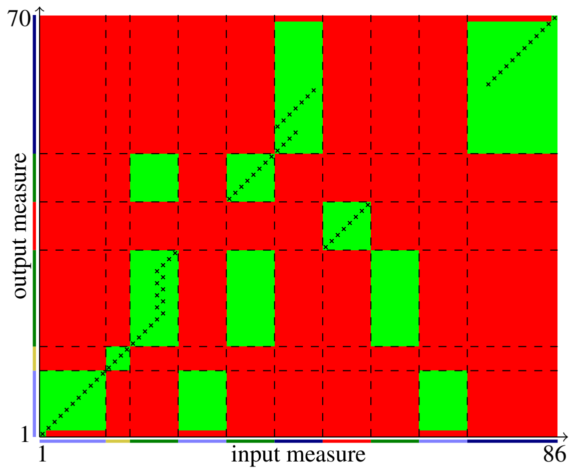

Figure 8: Constraint matrix for the song *Partyfreunde* by *Dave Sünti*, with input measures on the horizontal axis and output measures increasing from bottom to top. Green indicates the user-specified allowed regions (zero cost), red marks regions with a constant higher cost. The colored bars along the axes indicate the user-supplied semantic segmentation of the song (light and dark blue: chorus and variation, yellow: interlude, green: verse, red: bridge). The marked measures constitute the final result. Note how all user constraints are fulfilled.

listener would notice, we achieve an overall system performance that allows to be employed in the field. Advances in beat tracking may broaden the usage of this approach to a wider variety of musical genres like experimental or avant-garde as well as stabilize results on existing genres known to work rather well.

In our approach, constraints can be specified in the form of song-part annotations so that song parts may be removed, scaled or shifted around. In western popular music, the lengths of song parts are often small multiples of a fixed number of measures. In that case, acceptable transitions may not be available for all user-specified sets of constraints. In future work, we would like to provide interactive feedback on such potential problems during input of constraints, including appropriate suggestions about how the problems can be circumvented, e.g., by making the part lengths in the output a multiple of those in the input.

In the future, automatic recognition of musical structure [PMK10] could be employed to simplify the annotation process and to provide a more intuitive interface to our algorithm. Music structure analysis is still an active area of reserch, but even today existing methods may give an initial hint about a song's part distribution. Combined with our analysis and resynthesis algorithm, such methods could provide a valuable tool to aid video editors, dance choreographers and other content producers alike to produce music scaled to their desires.

References

[BC57] Bray J., Curtis J.: An ordination of the upland forest communities of Southern Wisconsin. *Ecological monographs 27*, 4 (1957), 325–349. 3

[BDA*05] Bello J., Daudet L., Abdallah S., Duxbury C., Davies M., Sandler M.: A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing 13*, 5 (2005), 1035–1047. 2

[BT59] Blackman R. B., Tukey J.: *The measurement of power spectra: from the point of view of communications engineering*, vol. 1058. Dover Publications New York, 1959. 3

[CBR03] Cardle M., Brooks S., Robinson P.: Audio and user directed sound synthesis. In *Proceedings of the International Computer Music Conference (ICMC), Singapore* (2003). 2

[Cha05] Chai W.: *Automated analysis of musical structure*. PhD thesis, Massachusetts Institute of Technology, 2005. 2

[Che02] Cheng K.: Beat this, a beat synchronization project. http://www.clear.rice.edu/elec301/Projects01/beat_sync/, 2002. 3

[CLSB06] Cannam C., Landone C., Sandler M., Bello J. P.: The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval* (2006), pp. 324–327. 3

[CT65] Cooley J., Tukey J.: An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation 19*, 90 (1965), 297–301. 4

[Dix01] Dixon S.: An empirical comparison of tempo trackers. In *Proceedings of the 8th Brazilian Symposium on Computer Music* (2001), pp. 832–840. 2

[Dix07] Dixon S.: Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research 36*, 1 (2007), 39–50. 3

[DP07] Davies M., Plumbley M.: Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing 15*, 3 (2007), 1009–1020. 2

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

[EKR87] Eckmann J., Kamphorst S., Ruelle D.: Recurrence plots of dynamical systems. *Europhysics Letters (EPL) 4* (1987), 973. 3

[Flo11] Flohrer T.: [auftakt] SDK 3.0.2 documentation. 2

[FM33] Fletcher H., Munson W. A.: Loudness, its definition, measurement and calculation. *The Journal of the Acoustical Society of America 5*, 2 (1933), 82–108. 3

[Foo99] Foote J.: Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)* (1999), ACM, pp. 77–80. 3

[GM94] Goto M., Muraoka Y.: A beat tracking system for acoustic signals of music. In *Proceedings of the second ACM international conference on Multimedia* (1994), ACM, pp. 365–372. 2

[KEA06] Klapuri A., Eronen A., Astola J.: Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing 14*, 1 (2006), 342–355. 2

[Lin] LINDOS ELECTRONICS: A-weighting in detail. http://en.wikipedia.org/wiki/File:Acoustic_weighting_curves_(1).svg. 4

[LLW*02] LU L., LI S., WENYIN L., ZHANG H.-J., MAO Y.: Audio textures. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2002), vol. 2, pp. II–1761–II–1764. 2

[LWZ04] LU L., WENYIN L., ZHANG H.: Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing 12*, 2 (2004), 156–167. 1, 2

[MMDK07] MCKINNEY M., MOELANTS D., DAVIES M., KLAPURI A.: Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research 36*, 1 (2007), 1–16. 2

[Ols72] OLSON H. F.: The measurement of loudness. *Audio Magazine* (1972), 18–22. 4

[PB04] PARKER J., BEHM B.: Creating audio textures by example: tiling and stitching. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)* (2004), vol. 4, IEEE, pp. 317–320. 2

[Pea82] PEARL J.: *Reverend Bayes on inference engines: A distributed hierarchical approach.* Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982. 4

[PMK10] PAULUS J., MÜLLER M., KLAPURI A.: State of the art report: Audio-based music structure analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference* (2010), pp. 625–36. 2, 7

[Ros92] ROSENTHAL D. F.: *Machine rhythm: Computer emulation of human rhythm perception.* PhD thesis, Massachusetts Institute of Technology, Dept. of Architecture, Cambridge, MA, USA, 1992. 1

[RP07] ROBERTSON A., PLUMBLEY M.: B-keeper: A beat-tracker for live performance. In *Proceedings of the 7th international conference on New interfaces for musical expression* (2007), ACM, pp. 234–237. 3

[S*00] SCHWARZ D., ET AL.: A system for data-driven concatenative sound synthesis. In *Digital Audio Effects (DAFx)* (2000), pp. 97–102. 2

[Sch06] SCHWARZ D.: Concatenative sound synthesis: The early years. *Journal of New Music Research 35*, 1 (2006), 3–22. 1, 2

[SDP09] STARK A., DAVIES M., PLUMBLEY M.: Real-time beat-synchronous analysis of musical audio. In *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy* (2009), pp. 299–304. 3

[SERlG06] STROBL G., ECKEL G., ROCCHESSO D., LE GRAZIE S.: Sound texture modeling: A survey. In *Proceedings of the 2006 Sound and Music Computing (SMC) International Conference* (2006), pp. 61–5. 2

[Stu04] STURM B.: Matconcat: an application for exploring concatenative sound synthesis using MATLAB. *Proceedings of DAFx04, Naples, Italy* (2004). 2

[TF03] TAPPEN M., FREEMAN W.: Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.* (2003), Ieee, pp. 900–906. 4

[WBSH*13] WENNER S., BAZIN J.-C., SORKINE-HORNUNG A., KIM C., GROSS M.: Scalable music: Automatic music retargeting and synthesis. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 345–354. 2

[WM11] WENGER S., MAGNOR M.: Constrained example-based audio synthesis. In *Proc. IEEE International Conference on Multimedia and Expo (ICME) 2011* (July 2011). 2

[WM12] WENGER S., MAGNOR M.: A genetic algorithm for audio retargeting. In *Proc. ACM Multimedia 2012* (June 2012), pp. 705–708. 2, 4

[YFW03] YEDIDIA J., FREEMAN W., WEISS Y.: Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium 8* (2003), 236–239. 4

[ZP01] ZILS A., PACHET F.: Musical mosaicing. In *Digital Audio Effects (DAFx)* (2001). 2

[Zpl13] ZPLANE.DEVELOPMENT: *[aufTAKT] V3 Tempo and Beat Tracking.* zplane.development GmbH & Co. KG / Katzbachstr. 21 / D-10965 Berlin, Germany, www.zplane.de, 2013. 3