EVENT-SYNCHRONOUS MUSIC ANALYSIS / SYNTHESIS

Tristan Jehan

Massachusetts Institute of Technology
Media Laboratory
tristan@media.mit.edu

ABSTRACT

This work presents a novel framework for *music synthesis*, based on the perceptual structure analysis of pre-existing musical signals, for example taken from a personal MP3 database. We raise the important issue of grounding music analysis on perception, and propose a bottom-up approach to music analysis, as well as modeling, and synthesis. A model of segmentation for polyphonic signals is described, and is qualitatively validated through several artifact-free music resynthesis experiments, e.g., reversing the ordering of sound events (notes), without reversing their waveforms. Then, a compact "timbre" structure analysis, and a method for song description in the form of an "audio DNA" sequence is presented. Finally, we propose novel applications, such as music cross-synthesis, or time-domain audio compression, enabled through simple sound similarity measures, and clustering.

1. INTRODUCTION

Music can be regarded as a highly complex acoustical and temporal signal, which unfolds through listening into a sequential organization of perceptual attributes. A structural hierarchy [1], which has been often studied in the frequency domain (i.e., relationship between notes, chords, or keys) and the time domain (i.e., beat, rhythmic grouping, patterns, macrostructures) demonstrate the intricate complexity and interrelationship between the components that make music. Few studies have proposed computational models on the organization of timbres in musical scenes. However, it was shown by Deliège [2] that listeners tend to prefer grouping rules based on timbre over other rules (i.e., melodic and temporal) and by Lerdahl in [3] that music structures could also be built up from timbre hierarchies.

Here we refer to *timbre* as the sonic "quality" of an auditory event, that distinguishes it from other events, invariantly of its change in pitch or loudness. From an auditory scene analysis point of view, by which humans build mental descriptions of complex auditory environment, an abrupt event is an important sound source separation cue. Auditory objects get first separated and identified on the basis of common dynamics and spectra. Then, features such as pitch and loudness are estimated [4]. Moreover, the clear separation of sound events in time makes music analysis and its representation easier than if we attempted to model audio and music all at once.

Segmentation has proven to be useful for a range of audio applications, such as automatic transcription [5], annotation [6], sound synthesis [7], or rhythm and beat analysis [8] [9]. Data-driven concatenative synthesis consists of generating audio sequences by juxtaposing small *units* of sound (e.g., 150 ms), so that the result best matches a usually longer *target* sound or phrase. The method was first developed as part of a *text-to-speech* (TTS) system, which exploits large databases of speech phonemes in order to reconstruct entire sentences [10].

Schwarz's *Caterpillar* system [7] aims at synthesizing *sounds* with the concatenation of musical audio signals. The units are segmented via alignment, annotated with a series of audio *descriptors*, and are selected from a large database with a constraint solving technique.

Zils and Pachet's *Musical Mosaicing* [11] aims at generating *music* with arbitrary samples. The music generation problem is seen as a constraint problem. The first application proposed composes with overlapping samples by applying an overall measure of concatenation quality, based on descriptor continuity, and a constraint solving approach for sample selection. The second application uses a *target* song as the overall set of constraints.

Lazier and Cook's *MoSievius* system [12] takes up the same idea, and allows for real-time interactive control over the mosaicing technique by fast *sound sieving*: a process of isolating subspaces as inspired by [13]. The user can choose input and output signal specifications in real time in order to generate an interactive audio mosaic. Fast time-stretching, pitch shifting, and k-nearest neighbor search is provided. An (optionally pitch-synchronous) overlap/add technique is used for synthesis.

Few or no audio examples with these systems were available. Lazier's source code is however freely available online. Finally, a real world example of actual music generated with small segments collected from pre-existing audio samples is among others, John Oswald's Plunderphonics project. He created a series of collage pieces by cutting and pasting samples by hand [14].

2. AUDITORY SPECTROGRAM

Let us start with a monophonic audio signal of arbitrary sound quality—since we are only concerned with the musical appreciation of the audio by a human listener, the signal may have been formerly compressed, filtered, or resampled—and any musical content—we have tested our program with excerpts taken from jazz, classical, funk, pop music, to speech, environmental sounds, or simple drum loops. The goal of our auditory spectrogram is to convert the time-domain waveform into a reduced, yet perceptually meaningful, time-frequency representation. We seek to remove the information that is the least critical to our hearing sensation while retaining the important parts, therefore reducing signal complexity without perceptual loss. An MP3 codec is a good example of application that exploits this principle for compression purposes. Our primary interest here is segmentation (see Section 3), therefore the process is being simplified.

First, we apply a standard STFT to obtain a regular spectro-

gram. Many window types and sizes have been tested, which did not really have a significant impact on the results. However, since we are mostly concerned with timing accuracy, we favor short windows (e.g., 12 ms Hanning), which we compute every 3 ms (i.e., every 128 samples at 44.1 KHz). The FFT is zero-padded up to 46 ms to gain additional interpolated frequency bins. We now calculate the power spectrum, and then group and convert resulting bins into 25 critical-bands according to a Bark scale—see equation (1). At low frequencies, critical bands show an almost constant width of about 100 Hz while at frequencies above 500 Hz, they show a bandwidth which is about 20% of the center frequency [15].

$$z(f) = 13 \cdot arctan(0.00076f) + 3.5 \cdot arctan\Big((f/7500)^2\Big) \quad (1)$$

A non-linear *spreading function* is calculated for every frequency band with equation (2) [16]. The function models *frequency masking* and may vary depending on the refinement of the model. More details can be found in [17].

$$SF(z) = (15.81 - i) + 7.5(z + 0.474) - (17.5 - i)\sqrt{1 + (z + 0.474)^2} \quad (2)$$

where

$$i = \min(5 \cdot |F(f)| \cdot BW(f), 2.0), \text{ and}$$

$$BW(f) = \begin{cases} 100 & \text{for } f < 500 \\ 0.2f & \text{for } f \geq 500 \end{cases}$$

Another perceptual phenomenon that we consider as well is *temporal masking*, and particularly post-masking. The envelope of each critical-band is convolved with a 200-ms half-Hanning (i.e., raised cosine) window. This stage induces smoothing of the spectrogram, while preserving attacks. The outcome merely approximates a "what-you-see-is-what-you-hear" type of spectrogram, meaning that the "just visible" in the time-frequency display (see Figure 1, frame 2) corresponds to the "just audible" in the underlying sound. The spectrogram is finally normalized to the range 0-1.

Among perceptual descriptors commonly exploited stands out *loudness*: the subjective judgment of the intensity of a sound. It can be approximated by the area below the masking curve. We can simply derive it from our spectrogram by adding the energy of each frequency band (see Figure 1, frame 3).

3. SEGMENTATION

Segmentation is the means by which we can divide the musical signal into smaller units of sound. When organized in a particular order, the sequence generates music. Since we are not concerned with sound source separation at this point, a segment may represent a rich and complex polyphonic sound, usually short.

We define a sound segment by its onset and offset boundaries. It is assumed perceptually "meaningful" if its timbre is consistent, i.e., it does not contain any noticeable abrupt changes. Typical segment onsets include abrupt loudness, pitch or timbre variations. All of these events translate naturally into an abrupt spectral variation in our auditory spectrogram.

First, we convert the spectrogram into an *event detection function*. It is obtained by first calculating the first-order difference

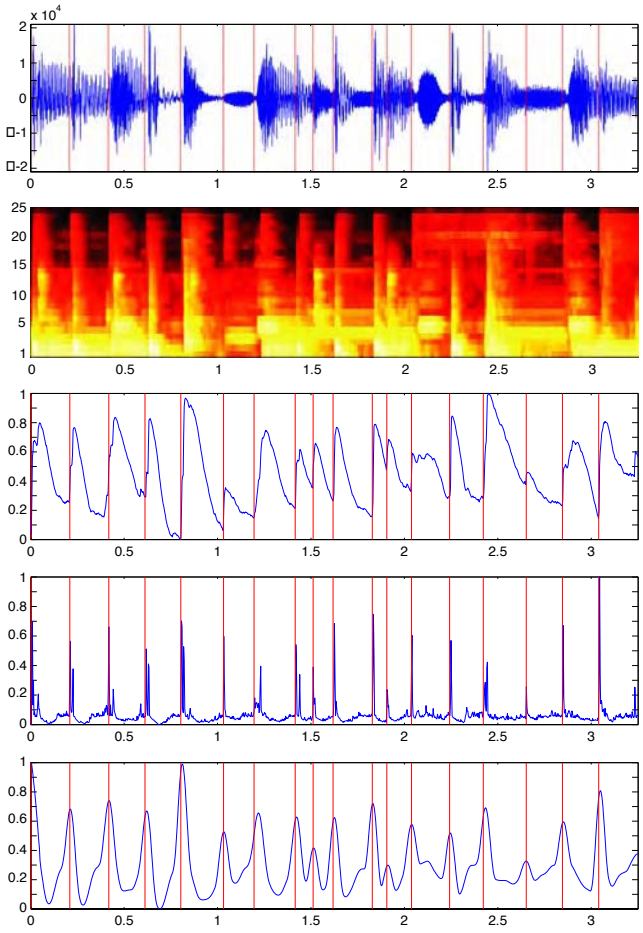

Figure 1: *A short 3.25 sec. excerpt of Watermelon man by Herbie Hancock. [from top to bottom] 1) the waveform (blue) and the segment onsets (red); 2) the auditory spectrogram; 3) the loudness function; 4) the event detection function; 5) the detection function convolved with a 150-ms Hanning window.*

function for each spectral band, and then by summing these envelopes across channels. The resulting signal contains peaks, which correspond to onset transients (see Figure 1 frame 4). We smooth that signal in order to eliminate irrelevant sub-transients (i.e., sub-peaks) which, within a 50 ms window would perceptually fuse together. That filtering stage is implemented by convolving the signal with a Hanning window (best results were obtained with a 150-ms window). This returns a smooth function, now appropriate for the *peak-picking* stage. The onset transients are found by extracting the local maxima in that function (see Figure 1, frame 5). A small arbitrary threshold could be necessary to avoid smallest undesired peaks, but its choice should not be critical.

Since we are concerned with reusing the audio segments for synthesis, we now refine the onset location by analyzing it in relationship with its corresponding *loudness function*. An onset would typically occur with an increase in loudness. To retain the entire attack, we search for the previous local minimum in that signal (i.e., usually a small shift of less than 20 ms), which corresponds to the softest moment before the onset (see Figure 1, frame 3). Finally,

we look in the corresponding waveform, and search for the closest zero-crossing, with an arbitrary but consistent choice of direction (e.g., negative to positive). This stage is important to insure signal continuity at synthesis (see Section 5).

4. BEAT TRACKING

Our beat-tracker was mostly inspired by Eric Scheirer's [18] and assumes no knowledge beforehand. For instance, it does not require a drum track, or a bass line to perform successfully. However, there are differences in the implementation which are worth mentioning. First, we use the auditory spectrogram as a front-end analysis technique, as opposed to a filterbank of six sixth-order elliptic filters, followed by envelope extraction. The signal to be processed is believed to be more perceptually grounded. We also use a large bank of comb filters as resonators, which we normalized by integrating the total energy possibly contained in the delay line, i.e., assuming DC signal. A *salience* parameter is added which allows us to estimate if there's a beat in the music at all. For avoiding tempo ambiguity (e.g., octaves), we use a template mechanism to select the faster beat, as it gives more resolution to the metric, and is easier to down-sample if needed.

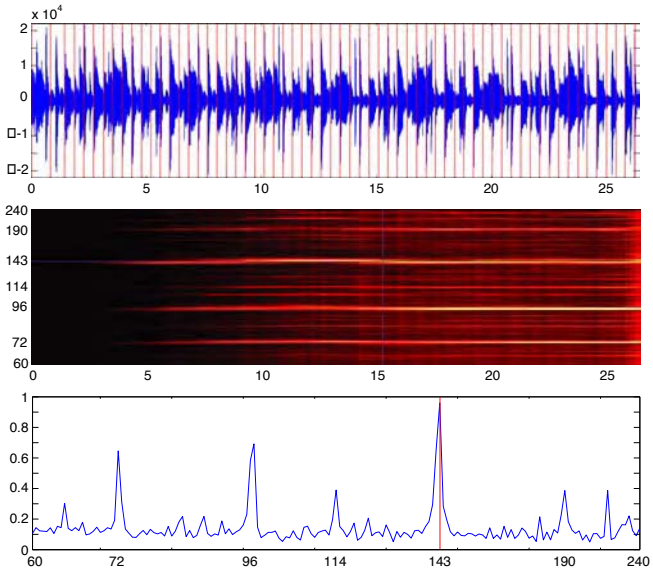

Figure 2: *Beat tracking of a 27 sec. excerpt of Watermelon man by Herbie Hancock. [from top to bottom] 1) the waveform (blue) and the beat markers (red); 2) the tempogram; 3) the tempo spectrum after 16 sec. of tracking.*

Figure 2 shows an example of beat tracking a polyphonic jazz-fusion piece at roughly 143 BPM. A tempogram (frame 2) displays the knowledge of tempo gained over the course of the analysis. First, there is no knowledge at all, but slowly the tempo gets clearer and stronger. Note in frame 1 that beat tracking was accurately stable after merely 1 second. The 3rd frame displays the output of each resonator. The strongest peak is the extracted tempo. A peak at the sub octave (72 BPM) is visible, as well as some other harmonics of the beat.

5. MUSIC SYNTHESIS

The motivation behind this preliminary analysis work is primarily synthesis. We are interested in composing with a database of sound segments—of variable sizes, typically ranging from 60 to 300 ms—which we can extract from a catalog of musical samples and pieces (e.g., an MP3 database), and which can be rearranged in a structured, and musically meaningful sequence, e.g., derived from the larger timbre, melodic, harmonic, and rhythmic structure analysis of an existing piece, or a specific musical model (another approach to combining segments could consist for instance of using generative algorithms).

In sound jargon, the procedure is known as *analysis-resynthesis*, and may often include an intermediary *transformation* stage. For example, a sound is analyzed through a STFT and decomposed in terms of its sinusoidal structure, i.e., a list of frequencies and amplitudes changing over time, which typically describes the harmonic content of a pitched sound. This represents the *analysis* stage. The list of parameters may first be transformed, e.g., transposed in frequency, or shifted in amplitude, and is finally resynthesized: a series of oscillators are tuned to each frequency and amplitude, and are summed to generate the waveform.

We extend the concept to "music" analysis and resynthesis, with structures derived from timbre which motivated the need for segmentation. A segment represents the largest unit of continuous timbre. We believe that each segment could very well be resynthesized by known techniques, such as additive synthesis, but we are only concerned with the issue of music synthesis, i.e., the structured juxtaposition of sounds over time, which implies higher level (symbolic) structures. Several qualitative experiments have been implemented, to demonstrate the advantages of a segment-based music synthesis approach over an indeed more generic, but still ill-defined frame-based approach.

5.1. Scrambled Music

This first of our series of experiments assumes no structure or constraint whatsoever. Our goal is to synthesize an audio stream by randomly juxtaposing short sound segments previously extracted from an existing piece of music—typically 2 to 8 segments per second with the music that was tested.

At segmentation, a list of pointers to audio segments is created. Scrambling the music consists of rearranging randomly the sequence of pointers, and of reconstructing the corresponding waveform. There is no segment overlap, windowing, or cross-fading involved, as generally the case with granular synthesis to avoid discontinuities. Here the audio signal is not being processed. Since segmentation was performed perceptually at a strategic location (i.e., just before an onset, at the locally quietest moment, and at zero-crossing), the transitions are artifact-free and seamless.

While the new sequencing generates the most unstructured music, the *event-synchronous synthesis* approach permitted us to avoid generation of audio clicks and glitches. This experiment is arguably regarded as the "worst" possible case of music resynthesis; yet the result is audiowise adequate to hearing (see Figure 3).

The underlying beat of the music, if any, represents a perceptual metric on which the segment structure fits. While beat tracking was found independently of the segment structure, the two representations are intricately interrelated with each other. The same scrambling procedure can be applied to the *beat segments* (i.e., audio segments separated by two beat markers).

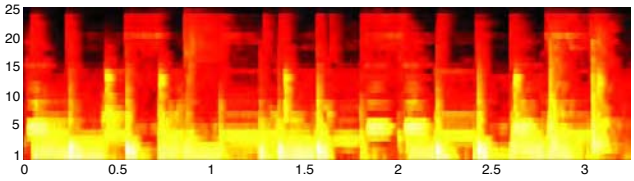

Figure 3: *Scrambled version of the musical excerpt of Figure 1.*

A new list of pointers to beat segments is created for the beat metric. If a beat marker occurs at less than 10% of the beat from a segment onset, we relocate the marker to that segment onset—strategically a better place. If there is no segment marker within that range, it is likely that there is no onset to be found, and we relocate the beat marker to the closest zero-crossing in order to minimize possible discontinuities. We could as well discard that beat marker altogether.

We apply the exact same scrambling procedure on that list of beat segments, and generate the new waveform. As predicted, the generated music is now metrically structured, i.e., the beat is found again, but the harmonic, or melodic structure are now scrambled. Compelling results were obtained with samples from polyphonic african, latin, funk, jazz, or pop music.

5.2. Reversed Music

The next experiment consists of adding simple structure to the previous method. This time, rather than scrambling the music, the segment order is entirely reversed, i.e., the last segment comes first, and the first segment comes last. This is much like what we could expect to hear when playing a score backwards, starting with the last note first, and ending with the first one. This is however very different from reversing the audio signal, which distorts the perception of the sound events since they start with an inverse decay, and end with an inverse attack (see Figure 4).

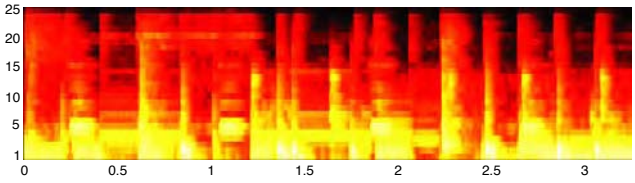

Figure 4: *Reversed version of the musical excerpt of Figure 1.*

The method has been tested successfully on several types of music including drum, bass, and saxophone solos, classical, jazz piano, polyphonic folk, pop, and funk music. It was found that perceptual issues with unprocessed reversed music occur with overlapping sustained sounds, or long reverb—some perceptual discontinuities cannot be avoided.

This experiment is a good test bench for our segmentation. If the segmentation failed to detect a perceptually relevant onset, the reversed synthesis would fail to play the event at its correct location. Likewise, if the segmentation detected irrelevant events, the reversed synthesis would sound unnecessarily granular.

The complete procedure, including segmentation and reordering, was run again on the reversed music. As predicted, the original piece was always recovered. Only little artifacts were encountered, usually due to a small time shift with the new segmentation, then resulting into slightly noticeable jitter and/or audio residues at resynthesis. Few re-segmentation errors were found. Finally, the reversed music procedure can easily be extended to the beat structure as well, and reverse the music while retaining a metrical structure.

5.3. Time-Axis Perceptual Redundancy Cancellation

A perceptual multidimensional scaling (MDS) of sound is a geometric model which allows the determination of the Euclidean space (with an appropriate number of dimensions) that describes the distances separating timbres as they correspond to listeners' judgments of relative dissimilarities. It was first exploited by Grey [19] who found that traditional monophonic pitched instruments could be represented in a three-dimensional timbre space with axes corresponding roughly to attack quality (temporal envelope), spectral flux (evolution of the spectral distribution over time), and brightness (spectral centroid).

Similarly, we seek to label our segments in a perceptually meaningful and compact, yet sufficient multidimensional space, in order to estimate their similarities in the timbral sense. Perceptually similar segments should cluster with each other and could therefore hold comparable labels. For instance, we could represent a song with a compact series of audio descriptors (much like a sort of "audio DNA") which would relate to the segment structure. Close patterns would be comparable numerically, (much like two protein sequences).

Thus far, we have only experimented with simple representations. More in-depth approaches to sound similarities and low level audio descriptors may be found in [20] or [21]. Our current representation describes sound segments with 30 normalized dimensions, 25 derived from the average amplitude of the 25 critical bands of the Bark decomposition, and 5 derived from the loudness envelope (i.e., loudness value at onset, maximum loudness value, location of the maximum loudness, loudness value at offset, length of the segment). The similarity between two segments is calculated with a least-square distance measure.

With popular music, sounds tend to repeat, whether they are digital copies of the same material (e.g., a drum loop), or simply musical repetitions with perceptually undistinguishable sound variations. In those cases, it can be appropriate to cluster sounds that are very similar. Strong clustering (i.e., small number of clusters compared with the number of original data points) is useful to describe a song with a small alphabet and consequently get a rough but compact structural representation, while more modest clustering (e.g., that is more concerned with perceptual dissimilarities), would only combine segment that are very similar with each other.

While modern lossy audio coders efficiently exploit the limited perception capacities of human hearing in the frequency domain [17], they do not take into account the perceptual redundancy of sounds in the time domain. We believe that by canceling such redundancy, we not only reach further compression rates, but since the additional reduction is of different nature, it would not affect "audio" quality per say. Indeed, with the proposed method, distortions if any, could only occur in the "music" domain, that is a quantization of "timbre", coded at the original bit rate. It is obviously arguable that musical distortion is always worse than audio distortion, however distortions if they actually exist (they would not if the sounds are digital copies), should remain perceptually undetectable.

We have experimented with redundancy cancellation, and obtained perfect resynthesis with simple cases. For example, if a drum beat (even complex and poly-instrumental) is looped more than 10 times, the sound file can easily be reduced down to 10% of its original size with no perceptual loss. More natural excerpts of a few bars were tested with as low as 30% of the original sound material, and promising results were obtained (see Figure 5). The more abundant the redundancies, the better the segment ratio, leading to higher compression rates. Our representation does not handle parametric synthesis yet (e.g., amplitude control), which could very much improve the results. Many examples were purposely over compressed in order to generate musical artifacts. These would often sound fine if the music was not known ahead of time. More on the topic can be found in [22].

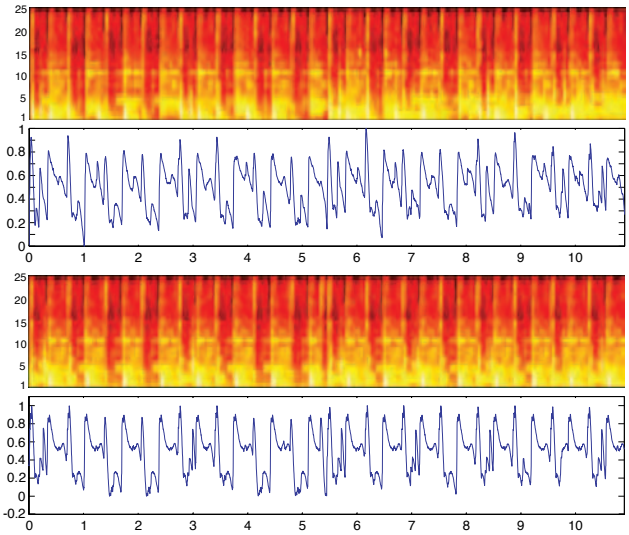

Figure 5: *[top] Original auditory spectrogram of 11 sec. of an african musical excerpt (guitar and percussion performed live), and its corresponding loudness function. [bottom] Resynthesized signal's auditory spectrogram with only 10% of the original material, and its corresponding loudness function.*

5.4. Cross-synthesis

Cross-synthesis is a technique used for sound production, whereby one parameter of a synthesis model is applied in conjunction with a different parameter of another synthesis model. Physical modeling, linear predictive coding, or the vocoder for instance enable cross-synthesis.

We extend the principle to the cross-synthesis of music, much like in [11], but we *event-synchronize*[1] segments at synthesis rather than using arbitrary segment lengths. We first generate a *source* database from the segmentation of a piece of music, and we replace all segments of a *target* piece by the most similar segments in the source. Each piece can be of arbitrary length and style.

The procedure relies essentially on the efficiency of the similarity measure between segments. Ours takes into account the frequency content as well as the time envelope, and performs fairly well with the samples we have tested. A more advanced technique based on dynamic programming is currently under development. We have experimented with cross-synthesizing pieces as dissimilar as a guitar piece with a drum beat, or a jazz piece with a pop song. Finally our implementation allows to combine clustering (Section 5.3) and cross-synthesis together—the target or the source can be pre-processed to contain fewer sounds, yet contrasting ones.

[1] the term here is given as an analogy with the term "pitch-synchronous," as found in PSOLA.

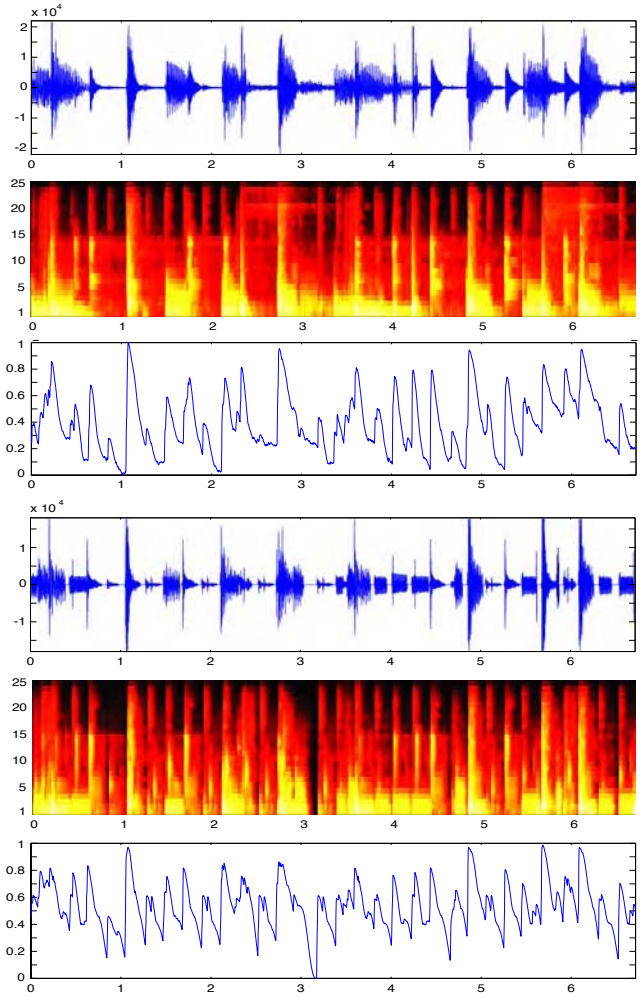

Figure 6: *Cross-Synthesis between an excerpt of Kickin' back by Patrice Rushen (source) and another excerpt of Watermelon man by Herbie Hancock (target). [top] The target waveform, its auditory spectrogram, and its loudness function. [bottom] The cross-synthesized waveform, its auditory spectrogram, and its loudness function. Note the close timing and spectral relationship between both pieces although they are made of different sounds.*

The results that we obtained were inspiring, and we believe they were due to the close interrelation of rhythm and spectral distribution between the target and the cross-synthesized piece. This interconnection was made possible by the means of synchronizing sound events (from segmentation) and similarities (see Figure 6).

Many sound examples for all the applications that were described in this paper, all using *default* parameters, are available at: `http://www.media.mit.edu/~tristan/DAFx04/`

6. IMPLEMENTATION

The several musical experiments described above easily run in a stand-alone Mac OS X application through a simple GUI. That application was implemented together with the *Skeleton* environment: a set of Obj-C/C libraries primarily designed to speed up, standardize, and simplify the development of new applications dealing with the analysis of musical signals. Grounded upon fundamentals of perception and learning, the framework consists of machine listening, and machine learning tools, supported by flexible data structures and fast visualizations. It is being developed as an alternative to more generic and slower tools such as Matlab, and currently includes a collection of classes for the manipulation of audio files (SndLib), FFT and convolutions (Apple's vDSP library), k-means, SVD, PCA, SVM, ANN (nodeLib), psychoacoustic models, perceptual descriptors (pitch, loudness, brightness, noisiness, beat, segmentation, etc.), an audio player, and fast and responsive openGL displays.

7. CONCLUSION

The work we have presented includes a framework for the structure analysis of music through the description of a sequence of sounds, which aims to serve as a re-synthesis model. The sequence relies on a perceptually grounded *segmentation* derived from the construction of an *auditory spectrogram*. The sequence is embedded within a *beat metric* also derived from the auditory spectrogram. We propose a clustering mechanism for time-axis redundancy cancellation, which applies well to applications such as audio *compression*, or timbre structure *quantization*. Finally, we qualitatively validated our various techniques through multiple synthesis examples, including reversing music, or cross-synthesizing two pieces in order to generate a new one. All these examples were generated with default settings, using a single Cocoa application that was developed with the auhor's *Skeleton* library for music signal analysis, modeling and synthesis. The conceptually simple method employed, and audio quality of the results obtained, attest for the importance of timbral structures with many types of music. Finally, the perceptually meaningful description technique showed clear advantages over brute-force frame-based approaches in recombining audio fragments into new sonically meaningful wholes.

8. REFERENCES

[1] Stephen McAdams, "Contributions of music to research on human auditory cognition," in *Thinking in Sound: the Cognitive Psychology of Human Audition*, pp. 146–198. Oxford University Press, 1993.

[2] I. Deliège, "Grouping Conditions in Listening to Music: An Approach to Lerdhal and Jackendoff's grouping preferences rules," *Music Perception*, vol. 4, pp. 325–360, 1987.

[3] F. Lerdhal, "Timbral hierarchies," *Contemporary Music Review*, vol. 2, pp. 135–160, 1987.

[4] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press, 1990.

[5] J. P. Bello, *Towards The Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach*, Ph.D. thesis, Queen Mary, University of London, 2003.

[6] George Tzanetakis and Perry Cook, "Multifeature audio segmentation for browsing and annotation," in *Proceedings IEEE Workshop on applications of Signal Processing to Audio and Acoustics*, October 1999.

[7] Diemo Schwarz, "The caterpillar system for data-driven concatenative sound synthesis," *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, September 2003.

[8] Christian Uhle and Juergen Herre, "Estimation of tempo, micro time and time signature from percussive music," in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, London, UK, September 2003.

[9] Masataka Goto, "An audio-based real-time beat tracking system for music with or without drum sounds," *Journal of New Music Research*, vol. 30, pp. 159–171, 2001.

[10] A. J. Hunt and A.W. Black, "Unit selection in a concatenative sound synthesis," in *Proceedings ICASSP*, Atlanta, GA, 1996.

[11] Aymeric Zils and Francois Pachet, "Musical mosaicing," in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-01)*, Limerick, Ireland, December 2001.

[12] Ari Lazier and Perry Cook, "Mosievius: Feature driven interactive audio mosaicing," *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, September 2003.

[13] George Tzanetakis, *Manipulation, Analysis, and Retrieval Systems for Audio Signals*, Ph.D. thesis, Princeton University, June 2002.

[14] John Oswald, "Plunderphonics' web site," 1999, `http://www.plunderphonics.com`.

[15] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer Verlag, Berlin, 2nd edition, 1999.

[16] T. Painter and A. Spanias, "A review of algorithms for perceptual audio coding of digital audio signals," 1997. Available from `http://www.eas.asu.edu/speech/ndtc/dsp97.ps`

[17] Marina Bosi and Richard E. Goldberg, *Introduction to Digital Audio Coding and Standards*, Kluwer Academic Publishers, Boston, December 2002.

[18] Eric Scheirer, "Tempo and beat analysis of acoustic musical signals," *Journal of the Acoustic Society of America*, vol. 103, no. 1, January 1998.

[19] J. Grey, "Timbre discrimination in musical patterns," *Journal of the Acoustical Society of America*, vol. 64, pp. 467–472, 1978.

[20] Keith Dana Martin, *Sound-Source Recognition. A Theory and Computational Model*, Ph.D. thesis, MIT Media Lab, 1999.

[21] Perfecto Herrera, Xavier Serra, and Geoffroy Peeters, "Audio descriptors and descriptor schemes in the context of MPEG-7," *International Computer Music Conference*, 1999.

[22] Tristan Jehan, "Perceptual segment clustering for music description and time-axis redundancy cancellation," in *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, Spain, October 2004.