Sound Art – Synthesis Based on Rhythm and Feature Extraction

Luka Mikula, Alois Sontacchi**, Robert Höldrich****

*University of Music and Dramatic Arts Graz, Austria, lukamikula@student.kug.ac.at
**Institute of Electronic Music and Acoustics Graz, Austria, sontacchi@iem.at
***Institute of Electronic Music and Acoustics Graz, Austria, hoeldrich@iem.at

Abstract

This paper presents a software tool that was developed for the transformation of audio signals, be it short sounds or whole songs. This can subsequently be used to synthesize music from audio signal fragments ("Concatenative Sound Synthesis"). It is attempted to use timbre-related features to re-synthesize audio signals. As in a mosaic, an existing song is newly constructed from small parts ("frames") of other songs already stored in a database. Suitable database frames are found by feature similarity analysis. The length of the frames corresponds to musically meaningful units. For the implementation suitable onset- and beat tracking methods are evaluated. The selection of suitable parameters describing the subjective semantic similarities is determined by listening tests.

1. Introduction

The process of putting together short recorded audio segments to form new signals has been often used as an artistic tool starting from the second half of the 20th century. Artists such as Pierre Schaeffer [1], John Cage [2] or Iannis Xenakis [3] manipulated short fragments of magnetic tape to create their compositions. Schaeffer presented a composition concept where short sound segments are defined as the "basic units of composition" [1]. This approach translated into the era of computers and digital sound processing, where composers like Curtis Roads [4], Barry Truax or John Oswald [5] use short samples of recorded sounds to create new audio material.

This paper describes a software tool that was developed for automatic synthesis of songs or sounds from an existing database of audio material. A combination of database segments that closely match the rhythmic and melodic structure of a *target song* selected by the user is concatenated to form a new song. The target song is segmented using onset detection to find the beats of the signal and consequently a musically meaningful division into short frames. The best-matching frames in the database are found by a similarity analysis of *low-level features*. The found frames are then transformed and concatenated to form the new signal.

There have been a number of approaches by other authors to find musically acceptable ways to re-synthesize songs from existing audio material. [5] uses a software tool that uses fixed segmentation lengths and lets the user choose from a number of options concerning the segment windowing and low-level feature comparisons. [6] created extension libraries for

SuperCollider where audio files are chopped up according to the found beats and the resulting frames are reassembled, creating a “jumbled” version of the original song. [7] synthesizes audio material corresponding to a specified MIDI score from an existing monophonic audio file. [8] segments two sounds by dividing them into units and calculating their distance; the pair that displays the largest distance is switched. [9] uses a loop-based approach where the user can choose feature ranges, and only segments whose features lie in the specified range are picked. This approach includes a selection mechanism that limits the segment search space by isolating sub-spaces that have desired elements in common. [10] aligns audio data with its symbolic score, where the algorithm tries to minimise cost functions that describe the dissimilarity between database and target song segments and the dissimilarity between successive re-synthesized frames.

This paper is organised as follows: chapter 2 presents the implemented re-synthesis interface. It describes the organisation of the sound material database and the work-flow of the algorithm, including the segmentation and the segment matching stages. The manner in which the found database segments are transformed and concatenated to form the re-synthesized song is also described briefly.

A number of different approaches to onset detection are implemented and evaluated in this paper. A functioning onset detection is crucial to provide meaningful segmentation points. Section 3 presents an overview of the implementations and the evaluation results.

In order to find the database segments that best match the target song segments, the similarity between these has to be described using parameters that can be understood by a computer. In order to find a combination of low-level features that best represents the perceived similarity between signals, a listening test was carried out. Its results are presented in section 4.

Section 5 gives an overview of important issues and problems that have been addressed in this paper. It also mentions possible future research topics that are crucial to producing musically and perceptively meaningful sound re-synthesis results.

2. Synthesis Algorithm Implementation

The re-synthesis algorithm that was implemented in Matlab is presented in this chapter. The necessary steps for the creation and organisation of an audio material database as well as for the re-synthesis of target songs are described in the sub-sections of this chapter.

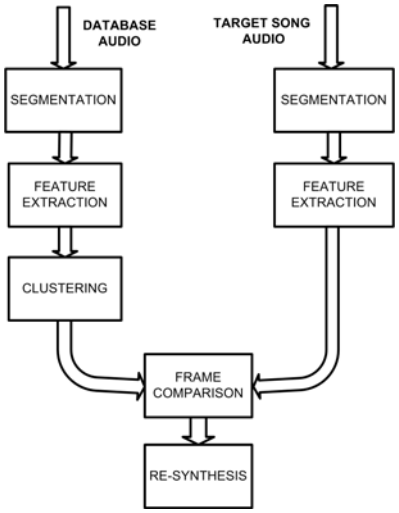

Figure 1: Re-synthesis algorithm work-flow

2.1. Segmentation

In order to divide audio material into musically meaningful regions, the boundaries between those regions have to be established. In our case, this is done by finding rhythmic events on the *tactus* metrical level, which is the level a listener would tap along to and which is commonly referred to as the "beat" of a signal [11]. The rhythmic events are found by creating an *onset detection function* from the audio material, which represents a sub-sampled version of the audio signal with maxima at the probable locations of events or changes [12]. The different approaches to creating a detection function that were implemented are described in detail in chapter 3.

The audio files are limited to one channel and down-sampled to 11.025 kHz. A *Short-Time Fourier Transform* (STFT) is calculated using a fixed frame length of 128 samples corresponding to 11.6 ms and a hop-size of 64 samples. The frame is weighted with a Hanning window and zero-padded to a length of 256 samples.

The next step consists of creating a detection function from the spectral data. To this end, four different approaches were evaluated which are explained in detail in section 3.

The resulting detection function is *smoothed* using a low-pass filter to remove noise and spurious peaks. In order to find only the positive changes which are indicative of onsets, the detection function is then differentiated and half-wave rectified.

$$df[m] = \frac{\left(\left|x[m] - x[m-1]\right| + \left(x[m] - x[m-1]\right)\right)}{2} \qquad (1)$$

where *df[m]* stands for the detection function and *m* for the frame number.

The locations of the onsets are then determined by picking out the maxima in the detection function. As a certain dynamic fluctuation is to be expected in many audio signals, an adaptive threshold is necessary. This threshold is computed by using a fixed threshold parameter in combination with an adaptive threshold parameter that is calculated by computing the local median of a moving window

$$thr[m] = \delta + \lambda \cdot median\left(df\left[m - \frac{H}{2} ... m + \frac{H}{2} \right] \right) \quad (2)$$

where *thr[m]* is the threshold curve, H the window length for the median computation, δ is the fixed threshold parameter and λ the weighting factor for the adaptive threshold. The threshold parameters should be chosen with care since they have considerable influence on the detection results. Any detection functions above this threshold are assumed to be onsets. Two additional processing stages aim to eliminate any false or multiple detections. The first one assumes that only actual peaks are onsets, i.e. the maximum has neighbouring values that are lower:

$$df[m-2] < df[m-1] < df[m] > df[m+1] > df[m+2] \quad (3)$$

The second eliminates multiple detections by using a window that has the length of a sixteenth note at 208 *beats per minute* (BPM), which is assumed to be the shortest musically meaningful metric unit. Only the point with the maximal detection function value in this window is selected.

The found onset locations are then passed on to an *inter-onset interval* (IOI) *beat tracker*. The time spans between onsets are evaluated, a procedure that is often found in literature ([15], [16], [17], [18]). This is done by creating histograms of the IOI distribution for two distinct metrical levels, the tactus or "beat" level and the *tatum* ("temporal atom") level, which describes the rate of the shortest meaningful pulse periods. The histograms are weighted with the *log-normal distribution* proposed by Parncutt [11]

$$p(\tau) = \frac{1}{\tau\sigma\sqrt{2\pi}} e^{\frac{1}{2\sigma^2}\left[\ln\left(\frac{\tau}{m}\right)\right]^2} \quad (4)$$

with τ as the IOI length in seconds and the shaping parameters σ and m which are set with fixed values of 0.28 and 0.55 for the tactus level and 0.39 and 0.18 for the tatum level. From these weighted histograms, beat hypotheses are constructed for both metrical levels by looking at the highest peaks in the histogram. The reliability of the hypotheses is first evaluated by comparing the arithmetic and the geometric mean. If the means show significant divergence, which indicates the presence of sharp peaks in the histogram, the hypotheses are assumed to be valid, whereas if the means are similar (the distribution of values is wider and the peaks are similarly high) the hypotheses are discarded. In the last step, the beat hypotheses for both metrical levels are compared in order to arrive at a single hypothesis which is then used to correct the previous onset detection results by eliminating any onsets that do not conform to the hypothesis.

2.2. Feature Extraction

After segmentation of the audio files, low-level features that describe spectral and temporal characteristics are computed for every segment. These can be combined into higher-level *descriptors* describing "the sound of the song" [19]. The rating of similarity between audio signals using a number of features is a widely-researched topic ([19], [20], [21], [22]), especially in the context of standard audio descriptions like the MPEG-7 standard [23].
On the assumption that most segments start with an "attack" phase during which the signal is not predictable, every segment is again divided into one region of fixed length (128 samples or 11.6 ms) – the "transient" region – and one segment with variable length that lies between the end of the transient region and the start of the next audio segment where the signal is assumed to be stable. Only this stable region is analysed in regard to descriptive features.
For similarity evaluation purposes, a six-dimensional feature vector is calculated for every segment. The elements of the feature vector were determined in the course of the listening test described in chapter 4 and consist of the following low-level features:

- The first *Mel-Frequency Cepstral Coefficient* (MFCC), which effectively describes the mean short-time energy changes in a signal[1];
- the *zero-crossing rate*, which is defined as the number of times a signal changes sign during a specified time span;
- the segment *pitch*
- and three statistical parameters describing the shape of the spectral distribution, the *skewness*, *kurtosis* and *flatness* parameters

2.3. Clustering

Two aims were pursued in the organisation of the audio material database: the search time should be minimal, and the sorting manner should mirror perceptual criteria. The solution of this problem was found in creating a three-dimensional look-up cube instead of a look-up table. The found audio segments are rearranged by sorting them into clusters and sub-clusters according to the segment length and the segment pitch. Since the length of the segment is directly related to the beat structure of the signal and the pitch is related to the spectral characteristics, this equals sorting the segments according to their rhythmic and melodic characteristics. Each resulting cluster/sub-cluster cube part is filled with the same number of segments, ensuring uniform distribution of audio segments over the cube.
This sorting mechanism leads to two consequences: first, the segments are stored in such a way that frames lie in close proximity to frames displaying similar rhythmic and melodic characteristics, while more different segments are separated by greater distances. This mirrors the neuro-physiologic organisation of the human and animal cortex [21]. Secondly, the search time for database segments matching the current target song segment is greatly reduced because not all database segments have to be evaluated, only the segments stored in the cluster/sub-cluster with similar length and pitch values have to be compared to the target song segment.

2.4. Frame Comparison

The distance or dissimilarity between database and target song segments is found by calculating the Euclidean distance between the respective six-dimensional feature vectors described in sub-section 2.3:

[1] In literature, the MFCCs are indexed starting from 0 or from 1. In this case, the second approach is used.

$$\Delta_s = \sqrt{\sum_{k=1}^{6} \left(x_k - y_k \right)^2} \tag{5}$$

where Δ_s is the segment dissimilarity (i.e. distance), *k* stands for the index of the feature, *x* for the database segment and *y* for the target song segment. The characteristics of the database segments that match the target song segments are stored in a look-up table for the following re-synthesis stage.

2.5. Re-Synthesis

The last algorithm stage consists of transforming the previously selected database segments and concatenating them to form a new audio signal. Since in most cases the database segments will have different lengths and different volume, they have to be transformed to ensure a continuous and stable re-synthesis result.
The volume adaptation of the database segment is achieved by using a gain curve that reaches the maximum gain only at a position well into the stable region of the segment, which leaves the transient region unchanged and only modifies the stable region.
For segment length consistency, two cases have to be distinguished: if the database segment is longer than the target song segment, it is simply cut; if the segment is too short, it is stretched to achieve the desired length. Stretching was found to be more fitting than looping the fragment because simple looping methods lead to perceived discontinuities in the signal and more complex looping approaches are computationally intensive. The stretching is performed by using the *TimeScale SOLA* algorithm presented in [24] which is based on the synchronous overlapping of segment regions.

3. Onset Detection

Onset detection is an important part of the sound re-synthesis algorithm presented in this paper because it ensures a perceptually meaningful segmentation of database and target song audio material. The need for reliable and fast onset detection systems arose with the advent of automatic music analysis and the areas where they are applied have increased considerably over the last few years and include harmonic analysis [13], database management and indexing [25] and signal transformations, including digital audio effects [26]. Existing onset detection systems in the time domain focus on the change in the amplitude or energy of the signal [12] or on the signal change in relation to the signal level [27]. In the frequency domain, some authors analyse the change between the energy of successive short time spectra ([28], [29]), others correlate short-time power spectra [25] or evaluate changes in the *complex frequency domain* [30]. Approaches using dyadic wavelet decomposition [31] and *Transient Modelling Synthesis* [32] have also been used. Another possibility of finding onset events is to use a probabilistic approach where the conformity of the audio signal to a signal model is evaluated [33].
This chapter describes the different approaches to creating a detection function that were evaluated; one is based on the approach described in [30] where changes of the Fourier coefficients in the complex frequency domain are evaluated (section 3.1), the second one tracks the energy in *chroma* or *pitch classes* (section 3.2), the third follows the first MFC coefficient over time (section 3.3), and the fourth approach looks at changes in the *modulation spectrum* of a signal (section 3.4). The evaluation results concerning these

methods are detailed in section 3.5. The pre-processing, post-processing, peak-picking and correction stages of the onset detection algorithm are described in section 2.1.

3.1. Onset Detection in the Complex Frequency Domain

In most cases, an onset event is accompanied by changes in the amplitude and phase spectrum of a signal. [30] uses this fact to find onsets by evaluating the change of Fourier coefficients in the complex frequency domain. This approach was implemented in order to be able to compare the onset detection methods implemented in this paper to an approach that proved to work well for a number of different signals [30].
The expected combination of spectral magnitude and phase for the *k*-th STFT bin is given by

$$\hat{S}_k[m] = \hat{R}_k[m] e^{j\hat{\phi}_k[m]} \tag{6}$$

where $\hat{R}_k[m]$ is the expected magnitude value and should, for stationary frames, equal the magnitude of the previous frame; and $\hat{\phi}_k[m]$ is the expected phase value. On the other hand, the actual spectral magnitude and phase for the *k*-th STFT bin is given by

$$S_k[m] = R_k[m] e^{j\phi_k[m]} \tag{7}$$

The measure for the stationarity for the *k*-th bin of a signal between two successive frames can be computed by calculating the (Euclidean) distance between the actual and the expected complex vectors:

$$\Gamma_k[m] = \sqrt{\left\{\left[\Re\left(\hat{S}_k[m]\right) - \Re\left(S_k[m]\right)\right]^2 + \left[\Im\left(\hat{S}_k[m]\right) - \Im\left(S_k[m]\right)\right]^2\right\}} \tag{8}$$

The detection function is then created by summing the differences for all *N* bins of an STFT frame:

$$df[m] = \sum_{k=1}^{N} \Gamma_k[m] \tag{9}$$

3.2. Chroma-based Onset Detection

The chroma scale consists of twelve distinct values or pitch classes corresponding to the twelve semitones without enharmonic equivalents represented in the circle of fifths. When two different tones are found to belong to one chroma value, this implies that the tones are one or more octaves apart from each other. The tracking of chroma values as a twelve-dimensional vector over time leads to a time-frequency representation of the signal called *Chromagram* or *Harmonic Pitch Class Profile* (HPCP) [34]. By mapping short-time signal spectra onto a chroma scale, the energy of the signal is compressed from the number of STFT bins used to 12, leading to a very compact representation of the signal. After transforming the signal into the frequency domain using the STFT, the energy of the STFT bins is mapped onto a chroma scale reaching from 41 Hz (the E0 tone) to 5 kHz. A chroma vector is created where the pitch class that contains the highest amount of energy for every STFT frame is stored. The ratio of energy present in the pitch class to the total energy contained in the STFT frame is calculated:

$$r[m] = \frac{E[m]}{E_c[m]} \tag{10}$$

with *E[m]* as the total STFT frame energy and $E_c[m]$ as the energy in the current pitch class of the chroma vector. This ratio is then differentiated and half-wave rectified as described in chapter 2.1 to create a detection function where only positive changes in the ratio are considered and the local maxima in this detection function are picked out to find the onset time locations.

3.3. MFCC-based Onset Detection

The idea behind this onset detection algorithm is to track MFC coefficients over time. The calculation of the coefficients is realised as follows: first, the magnitude spectrum of the signal is determined and is then filtered by a *Mel filter bank*, which is a group of triangular filters that fulfils the purpose of grouping together frequency components according to the Mel scale, which is based on the human perception of pitch distances. The logarithm is then computed over the summation of the frequency groups. This mirrors the behaviour of the human cochlea, where neuronal impulses are evaluated in frequency groups, resulting in an integration of the impulses. In the last stage of the calculation, the values obtained from the filter bank are transformed into the cepstral domain using the *Discrete Cosine Transform* (DCT) [35].

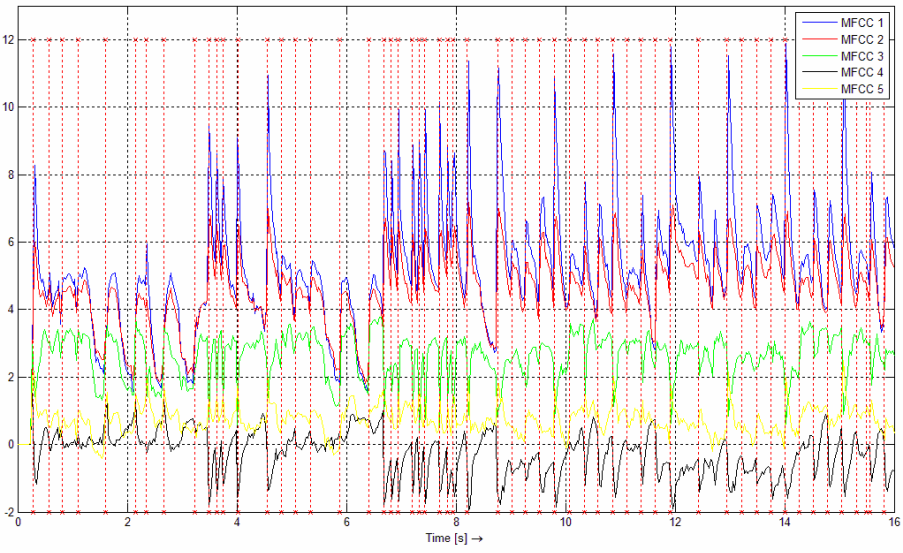

Figure 2: First 5 MFC coefficients and locations of hand-labelled onsets (red lines)

Fehler! Verweisquelle konnte nicht gefunden werden. shows the change in the first five MFC coefficients and onset positions determined by an expert listener. Using linear regression, a combination of weights was searched in order to create a weighted sum of the MFCCs to use as a detection function. However, these weights change considerably over different audio signals and genres, which is evidenced by **Fehler! Verweisquelle konnte nicht gefunden werden.**, which shows the different weighting factors determined for the first five MFCCs for a percussive signal (“Drums”), a non-percussive signal (“Fugees –

Ready Or Not") and a mixture of percussive and non-percussive sounds ("Cream – Sunshine Of Your Love").

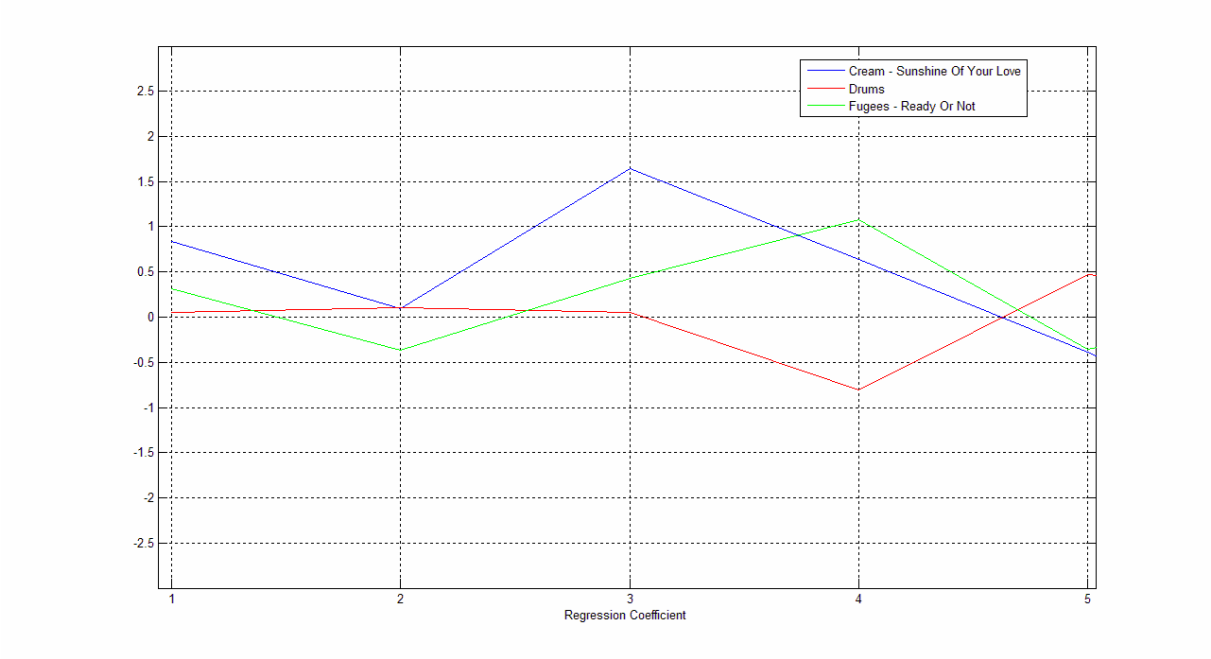

Figure 3: Weights for the first 5 MFC coefficients determined by linear regression for three different audio signals

As the weights visibly change with the audio signals, it was decided to track only the first MFCC over time since it has the sharpest peaks at onset locations (**Fehler! Verweisquelle konnte nicht gefunden werden.**). This can be explained by the fact that the first MFCC is a measure of energy change over time.

3.4. Onset Detection based on Modulation Spectra

Modulation spectra [14] are based on the interpretation of a STFT as the sub-band output of a filter bank. Every sub-band *trajectory*[1] can be analysed in regard to the changes in amplitude and phase spectrum by computing a second STFT. The obtained data representation is called modulation spectrum and is also known as the *modulation frequency domain*. Computing the STFT of $X_k[n]$[2] results in the transformed signal $\tilde{X}_k[p,q]$, the three-dimensional representation of the modulation spectrum with the modulation frequency p and the new time q. The time is decimated because the new time axis is determined by the original STFT frame rate.

The modulation spectrum domain is very useful for applications that concern themselves with evaluating the amplitude modulation of the respective sub-band trajectories. For example, a sinusoidal signal with constant amplitude will exhibit no changes in the modulation spectrum, there will only be a DC component visible. In contrast, any other signal with changing amplitude or frequency will exhibit clearly visible changes in the

[1] i.e. the temporal evolution of signal amplitude and phase in the frequency range determined by the sub-band center frequency and bandwidth

[2] The short-time spectrum of *x[n]* of the *k*-th sub-band

modulation spectrogram. The changes will be more or less pronounced according to the sub-band that is analysed. This property is exploited by evaluating the change in the modulation spectrum to find onsets in audio signals. The modulation spectrum itself is calculated as follows: first, a STFT of the signal is computed using a frame length of 128 samples (11.6 ms at 11.025 kHz) and a hop-size of 64 samples. The frame is weighted with a Hanning window and zero-padded to a length of 256 samples. The STFT is decomposed into 10 octave-spaced sub-bands, for every one of which a second STFT is computed using a frame length of 18 original STFT frames and a hop-size of 1. This leads to 10 modulation spectra, one for every sub-band. These are summed together using a weighting biased towards higher frequencies similar to the one used by [28], leading to one modulation spectrum representation.

$$\tilde{X}[q] = \sum_{p=1}^{10} p\tilde{X}_p[q] \tag{6}$$

where p stands for the sub-band index, $\tilde{X}_p[q]$ for the modulation spectrum of the p-th sub-band and $\tilde{X}[q]$ for the combined modulation spectrum representation. This weighting is used to exploit the fact that onsets are often accompanied by broad-band changes in the energy spectrum.

The weighted and summed modulation spectrum described above is divided into eight octave bands covering the complete modulation frequency bandwidth, in our case up to a little above 40 Hz. This is done because the energy in the low modulation frequencies tends to be uniformly high, in contrast to the higher modulation frequency bands where there are numerous sharp increases in the signal energy mainly due to onset events. A weighting similar to the method used to form the complete modulation spectrum presented previously is used:

$$df[q] = \sum_{p=1}^{8} p\tilde{X}_p[q] \tag{7}$$

where *df[q]* describes the detection function, p the modulation spectrum octave band index and $\tilde{X}_p[q]$ stands for the p-th modulation frequency octave band.

3.5. Evaluation

This chapter presents the results of the evaluation of the reduction algorithms presented in the previous sub-sections using a database of simple monophonic sounds and more complex pop songs from the last decades.

The audio files used for this purpose are all sampled at 44.1 kHz with 16-bit resolution. They are grouped into four categories: *non-pitched percussive* (NPP), *pitched percussive* (PP), *pitched non-percussive* (PNP) and *mixed* (M). The NPP files contain solely percussive sounds extracted from drum and sequencer tracks, the PP files contain songs using instruments such as bass guitars with clearly percussive attacks marking onsets, the PNP files are made up mostly from songs using soft synthesized sounds or instruments played *legato*, i.e. the transition between successive notes is smooth. Lastly, the sound files containing complex mixtures (M) are taken from pop songs. A total of 26 different sound

[1] i.e., no time decimation

files containing 1514 onsets is used in the evaluation process. The reference onsets are labelled by hand using the *Sound Onset Labeliser* interface from [36]. In order to fairly compare the different detection function implementations, the peak-picking algorithm using an adaptive threshold mentioned in section 2.1 is used throughout.
A detected onset is defined to be a correct detection if it falls within 35 milliseconds on either side of the according reference onset. While some authors use different tolerance regions for different signal types [37], in the interest of results compatibility across the databases a fixed tolerance time value is chosen. For comparison purposes, an evaluation score that combines the number of correctly detected onsets, false detections and missed detections is computed from the results [37]:

$$R = \frac{TP}{TP + FP + FN} \cdot 100\% \qquad (8)$$

where *TP* stands for the number of correct detections (true positives), *FP* for the number of incorrect detections (false positives) and *FN* for the number of missed detections (false negatives). The resulting *score ratio R* provides information about how well the onset detection algorithm works.

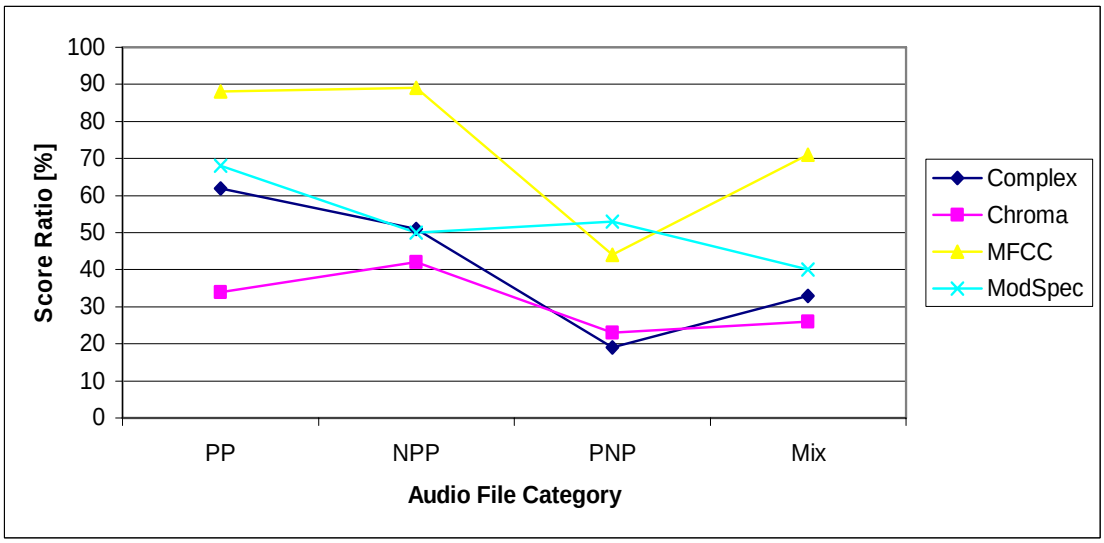

Figure 4: Detection results of the 4 implemented onset detection methods for different categories of audio signals

Fehler! Verweisquelle konnte nicht gefunden werden. shows the detection results of the previously presented algorithms for different signal categories. As was to be expected, the best results are achieved by all algorithms when faced with percussive signals. This is explained by the fact that all methods are in principle spectrum-based and percussive sounds lead to broad-band energy increases. The detection method based on following the first MFC coefficient over time works best for three out of four categories, which makes it the ideal candidate for onset detection tasks that have to deal with different signal types. The good performance can be explained by the fact that the MFCCs represent the spectral energy characteristic of signals in a very compressed way, which also explains the fact that the method does not perform as well with non-percussive signals where there are no significant energy changes. The approach using modulation spectra works best out of all methods when

faced with pitched non-percussive sounds, which is probably due to pronounced changes in the lower bands of the modulation spectrum.

4. Subjective Similarity Evaluation

A crucial task in any implementation of concatenative music synthesis is to find the audio segments that closely match the composer's vision of the overall sound. This is far from easy since the composer's concept of a sound may not match the actual physical parameters of the sound. This necessitates the characterisation of sound not by abstract physical parameters but by perceptually meaningful feature parameters extracted from the audio data. To gather more information about how listeners evaluate subjective similarity between different audio signals and to try to find a feature or a combination of features that best describes this perceived similarity, a listening test was carried out. From this test, the perceptual space that listeners use to evaluate similarities is analysed by using *Multi-Dimensional Scaling* (MDS), leading to a graphical representation of the perceived similarity distances. The results of the listening test are evaluated using the statistical software package SPSS.

The listening test was designed as a simple A-B comparison test, meaning the audio samples were played pair-wise. The task of the test subject was to define the subjective similarity between the two samples with a set of discrete values. A total number of 69 sample pairs was played back to the subjects, which for the most part can be classified as expert listeners, including a control pair that was repeated six times in a random order so as to check the reliability of the subjects' answers. The sample all had similar lengths (about 0.5 seconds) but displayed different melodic, harmonic and dynamic characteristics. The subjects' task was to rate the similarity between the samples on an eleven-grade scale ranging from very dissimilar to very similar. After normalising the answers by removing the mean and dividing by the variance for each test participant, the test results were evaluated using *non-metric* MDS.

In non-metric MDS, the order of proximities (in this case, dissimilarity distances) and not their order is meaningful, which corresponds to using an ordinal scaling method. The non-metric MDS representation is extracted from data by monotonic transformation of the object proximities. The points in an *n*-dimensional space are placed in such a way as to minimise the squared deviations between the scaled proximities and the distances between the points themselves. Mathematically, this requirement is described by the *stress factor*

$$s = \sqrt{\frac{\sum\left(f(\vec{p}) - \vec{d}\right)^2}{\sum \vec{d}^2}} \tag{9}$$

where s is the stress factor that has to be minimised, $\vec{p}$ is the proximity matrix, $f(\vec{p})$ stands for the monotonic transformation of $\vec{p}$ and $\vec{d}$ is the vector containing the distances between the points in the MDS representation. While there are different definitions of the stress factor available, SPSS uses the version defined by Kruskal [38] described above. The stress decreases when the number of dimensions is increased. Stress above 0.2 is considered an indication of a poorly-fitting solution while any number below 0.025 is judged to provide an excellent fit. The *proximity matrices* that describe the perceived (dis-)similarities between the audio samples were entered into the SPSS interface and analysed using the integrated MDS algorithm ALSCAL developed by Forrest Young [39]. This led to a two-

dimensional graphical representation of the audio sample relationships, which can be interpreted as the modelling of the perceptual space the subjects used to rate sample similarities. The stress factor for the two-dimensional solution lies at 0.005, indicating a very good quality of fit.

After evaluating the perceived distances between the samples pictured in **Fehler! Verweisquelle konnte nicht gefunden werden.** and the sound of the samples themselves, the two axes were defined as describing the *dynamic change* over time and the *spectral composition* of the sound. These loosely describe the temporal and spectral envelopes of the signals and are closely related to other evaluation dimension descriptions like *Log-Attack Time* or *Harmonic Spectral Centroid* and *Harmonic Spectral Spread* [22].

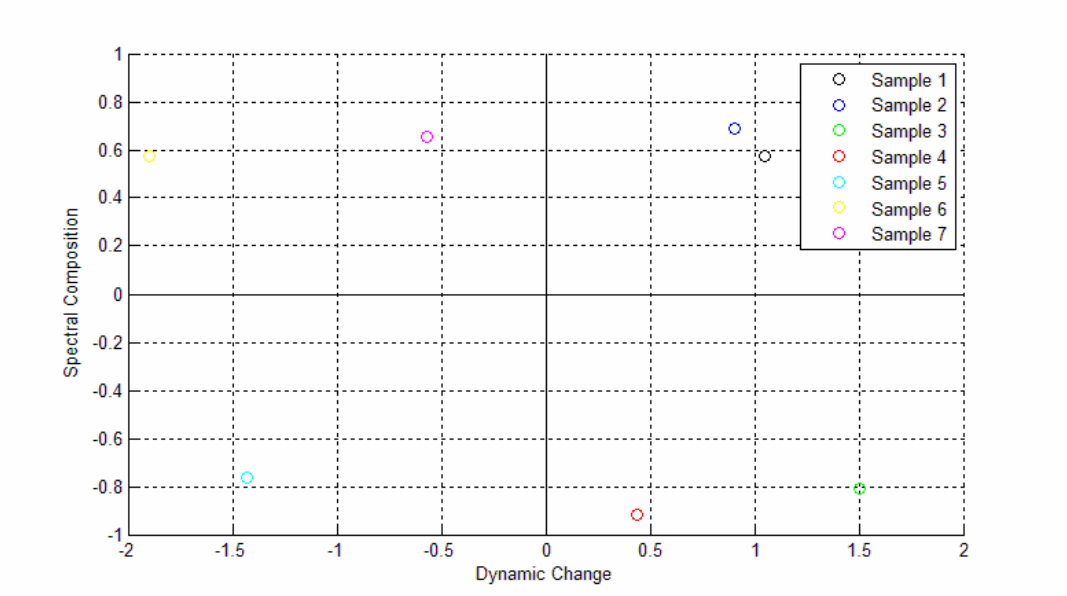

Figure 5: MDS representation of audio sample distances

The similarity distances found by the MDS evaluation of listening test answers were used to find a combination of low-level features that describes the signal as fully as possible while incorporating as few features as possible to reduce the computation time. To this end, over 40 different combinations of spectral, temporal and statistical features were evaluated. A matrix representation of an equation system is used to find the least-squares solution to the problem of finding the feature combination that yields the minimal residual between estimated and actual distances.
By checking the size of the residual for every solution, the combination of features that comes closest to describing the sample point distances with a minimal error is found. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows the ten best regression results. As was to be expected, a combination of all features worked best. The feature combination number 8 was found to have the best trade-off between exactness of results and computation complexity (the pitch calculation is already implemented in the re-synthesis tool for the database organisation) and is used for the frame matching algorithm stage described in section 2.2. The used features and their abbreviations are listed in appendix A.

	Feature Combination Name	Feats No.	Features	Residual
1	All	15	RMS, ZC, PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP, SK, KU, FL	0,006264923
2	Temporal / Spectral	12	RMS, ZC, PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP	0,045699485
3	Spectral / Stats	13	PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP, SK, KU, FL	0,092906633
4	Spectral	10	PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP	0,11985442
5	MFCCs / Energy / Stats	11	PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, IR, SP, SK, KU, FL	0,166007352
6	Tonality / Energy / Stats	13	PT, MFCC1, ZC, RO, CT, RN, RMS, BN, IR, SP, SK, KU, FL	0,178389407
7	MFCC1 / Centroid / Zerocross / Stats	6	MFCC1, CT, ZC, SK, KU, FL	0,187361823
8	MFCC1 / Pitch / Zerocross / Stats	6	MFCC1, PT, ZC, SK, KU, FL	0,189502726
9	MFCCs / Energy	8	RMS, MFCC1, MFCC2, MFCC3, RO, CT, BN, SP	0,22390577
10	Pitch	6	PT, MFCC1, RO, CT, BN, SP	0,243138767

Table 1: Best 10 regression results for different combinations of low-level features

5. Discussion of Results and Perspective

This paper presents a new approach to the re-synthesizing of audio signals from a database of existing audio material. A re-synthesis interface was implemented in Matlab to give the user control over different aspects of the synthesis algorithm. The algorithm uses a pre-defined database of audio signals for the re-synthesis of target songs. These target songs are analysed in regard to their beat structure so as to achieve musically meaningful segmentation results and the resulting segments are replaced by database segments using a similarity distance measure.

While evaluation of the re-synthesis results “by ear” show that the overall algorithm is far from perfect, it can be viewed as an encouraging step into the direction of musically and artistically valid algorithmic sound synthesis. It could be developed as a powerful tool for electronic music composition, where similar approaches have been introduced in recent years. It could eliminate the time-consuming and tiresome process of searching for sound sample material that corresponds to the artist’s vision – he or she could enter an exemplary sample and have the algorithm search for a matching sound segment.

This approach could also be integrated into process-based music and performances, allowing the artist to focus on sound combination and processing issues while the sound material is chosen algorithmically.

The algorithm could also be modified to create a recommendation system where users can input an exemplary sound sample, a chorus from a song or a whole song and the algorithm

could search for similar audio material in a database. This means that the segmentation which was implemented based on changes on the beat level of the signal would have to concentrate on higher-order metrical levels like measures or harmonic structures. A similar approach has been presented by [40] on the chorus level to create short, representative "thumbnails" out of music signals.

One area where the algorithm does not produce acceptable results is when music or audio is superimposed with vocals. For such cases, it will be necessary to find a way to re-synthesize the musical and the vocal parts of the signal separately.

As described in section 2.3, the database where segments are stored for later re-synthesis is designed as a three-dimensional cubic structure where segments are placed according to their length and their pitch value. This organisation greatly reduces computation time and ensures optimal usage of disk space, which is an issue with large sound databases. A possible approach to future database organisation could be to use not the uniform distribution of frames across clusters and sub-clusters as presented here but to model the cluster size according to the probability distributions of the frame data. This means that frequent length and pitch values are "quantised" using smaller steps.

One of the crucial issues of music re-synthesis is onset detection because only a well-functioning onset detection algorithm guarantees meaningful segmentation of audio signals. As **Fehler! Verweisquelle konnte nicht gefunden werden.** or [12] show, different methods work well only in specific contexts or for certain signal types. In the future, a method that delivers acceptable results over all possible audio signals is needed. The onset detection algorithm based on MFCCs presented in section 3.3 works well for three out of four signal categories, which is acceptable in most cases. Another possible approach could be to use a "modular" approach where the signal is first analysed in regard to percussiveness or genre (there have been attempts to extract the genre from music signal, see e.g. [41]) and an onset detection method suitable for this signal type is used.

Another important problem is the extraction of features from audio signals. In the context of database management and audio signal description[1], features have become an important issue in audio applications. Research topics in this field include instrument sound description [22], audio classification ([20], [42]) and the correlation between features [19]. Future tasks will include deeper research into the correlation between human perception and signal similarity. The relationship between subjective attributes such as "brightness", "sharpness" or "compactness" [21] and objective signal parameters such as "attack time" or "spectral deviation" [22] has not yet been fully explained. The main issue will be finding the smallest possible feature set that best describes the character of an audio signal.

Listening to the re-synthesis results shows that the principal structure of the target song is reproduced quite well, while melodic and rhythmic details and finer structures are not. Some of the unsolved issues and problems concerning music re-synthesis are addressed above, among them the need for improved onset detection and feature comparison, which in the authors' opinion are the most important unsolved problems. In conclusion, it can be said that this paper represents a step towards musically acceptable concatenative re-synthesis of audio signals, but there is still a lot of work to be done.

[1] the MPEG-7 standard defines such a standard for audio signal description [31]

6. Appendix A – Used Low-Level Features

RMS	Root Mean Square Energy		BN	Brightness
ZC	Zero-Crossing Rate		RN	Roughness
PT	Pitch		IR	Irregularity
MFCC1	First MFC Coefficient		SP	Spectral Spread
MFCC2	Second MFC Coefficient		SK	Skewness
MFCC3	Third MFC Coefficient		KU	Kurtosis
RO	Spectral Roll-Off		FL	Flatness
CT	Spectral Centroid			

7. Appendix B – References

[1] Schwarz D.: “Concatenative Sound Synthesis – The Early Years”, Journal of New Music Research 35 (1), 2006
[2] Kostelanetz R.: “John Cage”, M. DuMont Schauberg, Köln 1973
[3] Harley J.: “Xenakis – His Life in Music”, Routledge, New York 2004
[4] Roads C.: “Microsound”, MIT Press, Cambridge 2004
[5] Sturm B.: “Adaptive Concatenative Sound Synthesis and its Application to Micromontage Composition”, Computer Music Journal 30 (4), 2006
[6] Collins N.: “BBCut2: Integrating Beat Tracking and On-the-fly Event Analysis”, Journal of New Music Research 35 (1), 2006
[7] Simon I., Basu S., Salesin D., Agrawala M.: “Audio Analogies: Creating New Music from an Existing Performance by Concatenative Synthesis”, Proceedings of the International Computer Music Conference, Barcelona 2005
[8] Hazel S.: Soundmosaic Website, http://awesame.org/soundmosaic/, accessed April 14, 2008
[9] Lazier A., Cook P.: “MoSievius: Feature Driven Interactive Audio Mosaicing”, Proceedings of the Conference on Digital Audio Effects, London 2003
[10] Schwarz D.: “A System for Data-Driven Concatenative Sound Synthesis”, Proceedings of the Conference on Digital Audio Effects, Verona 2000
[11] Klapuri A., Eronen A., Astola J.: “Analysis of the Meter of Acoustic Music Signals”, IEEE Transactions on Speech, Audio and Signal Processing 14 (1), 2006
[12] 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
[13] Bello J., Pickens J.: “A Robust Mid-Level Representation for Harmonic Content in Music Signals”, Proceedings of the 6th International Symposium on Music Information Retrieval, London 2005
[14] Goodwin M., Avendano C.: “Frequency-Domain Algorithms for Audio Signal Enhancement Based on Transient Modification”, AES Journal 54 (9), 2006
[15] Dixon S.: “Automatic Extraction of Tempo and Beat from Expressive Performances”, Journal of New Music Research 30 (1), 2001

[16] Goto M., Muraoka Y.: "Music Understanding at the Beat Level – Real-Time Beat Tracking for Audio Signals", Working Notes of the IJCAI-95 Workshop on Computational Auditory Scene Analysis, Montréal 1995

[17] Goto M., Muraoka Y.: "Real-Time Rhythm Tracking for Drumless Audio Signals – Chord Change Detection for Musical Decisions", Working Notes of the IJCAI-97 Workshop on Computational Auditory Scene Analysis, Nagoya 1997

[18] Gouyon F., Herrera P.: "Determination of the Meter of Musical Audio Signals: Seeking Recurrences in Beat Segment Descriptors", Proceedings of the 114th AES Convention, Amsterdam 2003

[19] Mörchen F., Ultsch A., Thies M., Löhken I., Nöcker M., Stamm C., Efthymiou N., Kümmerer M.: "MusicMiner: Visualising Timbre Distances of Music as Topographical Maps", Technical Report No. 47, University of Marburg 2005

[20] West K., Cox S.: "Features and Classifiers for the Automatic Classification of Musical Audio Signals", Proceedings of the 5th International Symposium on Music Information Retrieval, Barcelona 2004

[21] Feiten B., Günzel S.: "A Sound-Retrieval Index Based on Two-Dimensional Similarity Maps", Proceedings of the 94th AES Convention, Berlin 1993

[22] Peeters G., McAdams St., Herrera P.: "Instrument Sound Description in the Context of MPEG-7", Proceedings of the International Computer Music Conference Berlin 2000

[23] MPEG-7 ISO Standard, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, accessed July 30, 2008

[24] Zölzer U., "DAFX: Digital Audio Effects", John Wiley & Sons, Chichester 2002

[25] Foote J.: "Automatic Audio Segmentation Using a Measure of Audio Novelty", IEEE International Conference on Multimedia and Expo, New York 2000

[26] Verfaille V., Arfib D.: "A-DAFX: Adaptive Digital Audio Effects", Proceedings of the Conference on Digital Audio Effects, Limerick 2001

[27] Klapuri A.: "Sound Onset Detection by Applying Psychoacoustic Knowledge", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Phoenix 1999

[28] Masri P.: "Computer Modelling of Sound for Transformation and Synthesis of Musical Signals", Ph.D. Dissertation, University of Bristol 1996

[29] Duxbury C., Sandler M., Davies M.: "A Hybrid Approach to Musical Onset Note Detection", Proceedings of the Conference on Digital Audio Effects, Hamburg 2002

[30] Bello J., Duxbury C., Davies M., Sandler M.: "On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain", IEEE Signal Processing Letters 11 (6), 2004

[31] Daudet L.: "Transients Modelling By Pruned Wavelet Trees", Proceedings of the International Computer Music Conference, Havana 2001

[32] Verma T., Levine S., Meng T.: "Transient Modeling Synthesis: A Flexible Analysis/Synthesis Tool for Transient Signals", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle 1998

[33] Abdallah S., Plumbley M.: "Probability As Metadata: Event Detection in Music Using ICA As Conditional Density Model", Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara 2003

[34] Harte C.r, Sandler M.: „Automatic Chord Identification Using a Quantised Chromagram", Proceedings of the 118th AES Convention, Barcelona 2005

[35] Sigurdsson S., Petersen K., Lehn-Schioler T.: "Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music", Proceedings of the 7th International Symposium on Music Information Retrieval, Victoria 2006

[36] Leveau P., "Sound Onset Labeliser", downloaded from www.lam.jussieu.fr/src/Members/Leveau/SOL/SOL.htm#DL, accessed November 7, 2007

[37] Collins N.: "A Comparison of Sound Onset Detection Algorithms with Emphasis on Psycho-acoustically Motivated Detection Functions", Proceedings of the 118th AES Convention, Barcelona 2005

[38] Wickelmaier F.: "An Introduction to MDS", Reports from the Sound Quality Research Unit, Aalborg University 2003

[39] Young F., ALSCAL Site, http://forrest.psych.unc.edu/research/alscal.html, accessed May 29, 2008

[40] Bartsch M., Wakefield G.: "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing", Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz 2001

[41] Tzanetakis G., Essl G., Cook P.: "Automatic Musical Genre Classification of Audio Signals", Proceedings of the 2nd International Symposium on Music Information Retrieval, Indiana 2001

[42] Bagci U., Erzin E.: "Automatic Classification of Musical Genres Using Inter-Genre Similarity", IEEE Signal Processing Letters, Volume 14 Issue 8, 2007