PERCEPTUALLY MOTIVATED PARAMETRIC REPRESENTATION FOR HARMONIC SOUNDS FOR DATA COMPRESSION PURPOSES

Marko Helen, Tuomas Virtanen

Institute of Signal Processing, Tampere University of Technology
P.O.Box 553, FIN-33101 Tampere, Finland
marko.helen@tut.fi

ABSTRACT

An efficient representation for the amplitude spectrum of harmonic sounds is proposed. The representation is based on modeling the rough spectral shape using Mel-frequency cepstral coefficients and their temporal evolution using the attack-decay-sustain-release-model. The representation significantly reduces the number of parameters while preserving most important perceptual features of sound. The proposed representation is applied as a part of an object-based audio coding system. Demonstrations of monophonic signals are available at http://www.cs.tut.fi/~heln/demopage.html.

1. INTRODUCTION

Sinusoidal modeling is widely used in audio signal processing. It aims at representing the periodic components of a signal as a sum of sinusoids, whose amplitudes, frequencies, and phases are time-varying.[1]

A harmonic sound has several features which allow further data compression: in case of a perfectly harmonic sound the frequencies of the overtones are integer multiples of the fundamental frequency (f0) so that only the f0 needs to be encoded. The phases are not perceptually important, and in many cases random phase or integral of frequency over time can be used.

The time-frequency spectrum of natural sounds is usually smooth which means that the amplitudes of harmonic components are slowly-varying. This has been utilized for example by using prediction and coding the prediction error [2]. In audio coding the temporal evolution of signal of it’s subchannels has been taken into account by using a gain slope, which for example reduces the pre-echo problems caused by quantization in frequency domain. For example, “Harmonic and Individual Lines plus Noise” (HILN) parametric coder [3] used intraframe gain slope for individual sinusoids.

In this paper, a method is proposed for the efficient parameterization of the amplitude spectrum of harmonic sounds. The method is based on two techniques which are widely used in audio applications but not in audio coding: Mel-frequency cepstral coefficients (MFCCs) are used to represent the spectrum of sounds and attack-decay-sustain-release (ADSR) -scheme is used to model the temporal evolution of parameters. Both techniques reduce significantly the amount of parameters while preserving the perceptually most important features, making the representation ideal for audio coding.

2. REPRESENTATION FOR HARMONIC SOUNDS

The harmonic part of a sound can be represented as a sum of sinusoids, so that $s_k(t)$, the k^{th} frame of sound can be expressed as:

$$s_k(t) = \sum_{n=1}^{N} a_{k,n} \cos(2\pi t\omega_{k,n} + \varphi_{k,n}) \quad , \tag{1}$$

where $a_{k,n}$, $\omega_{k,n}$ and $\varphi_{k,n}$ are the amplitude, frequency and phase of the n^{th} harmonic, respectively. Similarly, the harmonic sound $h_k(t)$ in one frame can be represented as a sum of harmonic partials

$$h_k(t) = \sum_{n=1}^{N} a_{k,n} \cos(2\pi tn\omega_{k,1} + \varphi_{k,n}) \quad , \tag{2}$$

where N is the number of harmonics.

The following section describes how the amplitudes $a_{k,n}$ can be efficiently parametrized.

2.1. Mel-cepstral representation

MFCCs are one of the best features used for example in instrument recognition. This suggest that the MFCCs represent the most important information of sound spectra from human sound perception point of view.[4]

As an input the algorithm takes amplitudes of every harmonic component of sound in every frame. For each frame the algorithm returns m cepstral coefficient where m is smaller than the number of harmonic components.

The MFCCs are calculated using the following procedure: Firstly, the amplitudes of the harmonic components are multiplied using a frequency-dependent weights which simulate the response of human auditory system. Secondly, a filterbank consisting of triangular filters spaced uniformly across the Mel-frequency scale [5] is simulated. Thirdly, the harmonic sum within each band is calculated. This can be formulated by

$$m_j = \sum_{n=1}^{N} W(\omega_n) H_j(\omega_n) \cdot a_n \quad , \tag{3}$$

where m_{j} is the harmonic sum in the j^{th} band, N is the number of harmonic components, $W(\omega)$ is the frequency-dependent weight at frequency ω. $H_j(\omega)$ is the frequency response of the j^{th} filter at

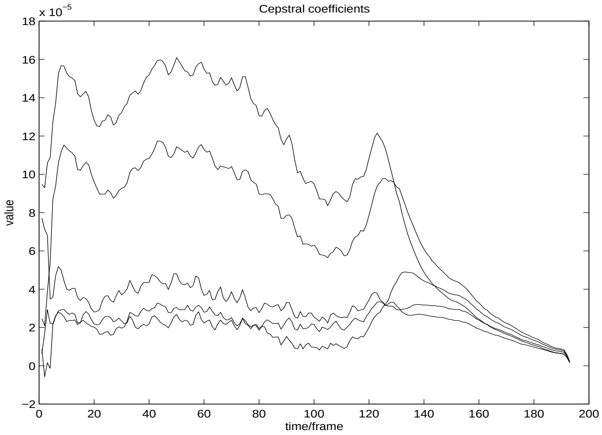

Figure 1: *First five cepstral coefficients of a bowed cello signal as a function of time.*

frequency ω, ω_n is the frequency of n^{th} harmonic component, and a_n is the amplitude of the harmonic component. Discrete cosine transform (DCT) is taken from m_j to get the cepstral coefficients

$$c_{mel}(i) = \sum_{j=1}^{N} m_j \cos\left(\frac{\pi}{N}\left(j - \frac{1}{2}\right)\right). \quad (4)$$

Finally, the first $M<N$ coefficients are selected to represent the rough spectral shape. An example of the MFCCs of a bowed cello are illustrated in Figure 1.

In the synthesis stage the MFCCs are transformed back to amplitudes. This reverse operation can be done with the first M coefficients. Inverse discrete cosine transform (IDCT) is taken from the M cepstral coefficients to approximate the energies within each band. The rest of the MFCCs are set to zero. The energies of each band are being divided uniformly along each frequency band. Finally, the effect of the auditory filter is compensated by a division. This procedure maintains the rough shape of spectrum and loses the details since the highest cepstral coefficients are discarded. In Figure 2 it can be seen that the generated amplitude spectrum is quite similar compared with the original one though the number of cepstral coefficients is only half of the harmonic components.

2.2. Temporal representation

ADSR is widely used in the sound synthesizers, but to our knowledge it has not been used in audio coding. The idea is that a continuous curve can be simplified using a series of line segments. The temporal evolution of a parameter is modeled with time/amplitude pairs, called start of attack, end of attack, start of release and end of release. An example of ADSR slope is illustrated in Figure 3.

Naturally, a problem is estimating the time/amplitude pairs from frame-wise amplitudes in a such manner that the sound synthesized using the parameters is perceptually as close to the original one as possible. The method proposed by Jensen [6] is used. However, the representation proposed in this paper does not specify the estimation method, so any other method can be used as well. Jensen's method consists of finding the maximum of the curve to be modeled, and then finding the first and the last time in the curve where the value is above a constant percent of the maximum. Only the maximum value and the instants of time have to be stored, since the rest of the values can be backtracked when knowing the percents. The percents, which are fixed before estimation, can be for example 10%, 90% and 70% so that the start of attack time is the first time the amplitude is above 10% of maximum, the end of attack time is the last time the amplitude is above 90%, the start of release is the last time the amplitude is above 70% and the end of release is the last time the amplitude is above 10% of the maximum [6]. An example of an original amplitude slope and the corresponding ADSR slope is plotted in Figure 3.

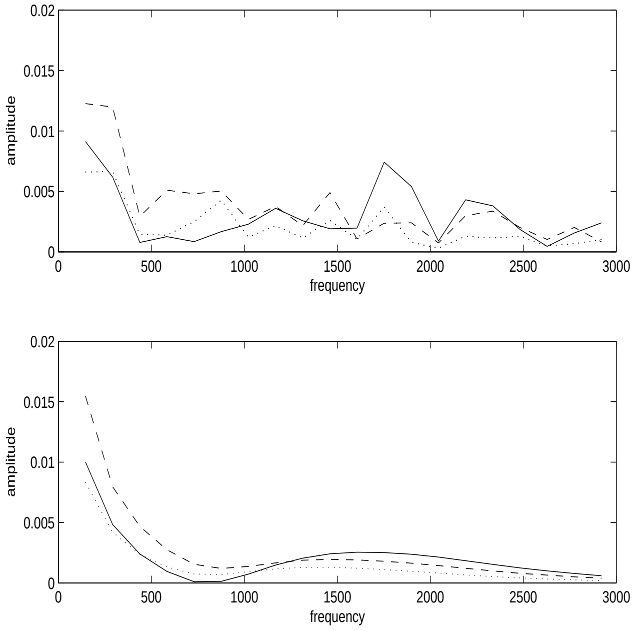

Figure 2: *Original sound and sound generated from cepstral coefficients at frames 1, 50 and 100 presented with solid, dashed and dotted line, respectively.*

In the basic form of ADSR the continuous curve between parameters is formed using linear or exponential pieces. A more flexible way is to add one parameter per line segment which parametrizes the curve between exponential, linear and logarithmic form. In our system, the parametrized curve is

$$C_s(x) = v_0 + (v_1 - v_0)(1 - (1 - x)^n)^{\frac{1}{n}}. \quad (5)$$

v_0 is the starting point, v_1 is the ending point, x is time parameter which is normalized between zero and one and $n>0$ is the adjustable parameter. When n is close to zero, the curve is exponential, when n is one, the curve is linear and when n is greater than one, the curve is logarithmic [6]. In our system, parameter n is found by minimizing the least-square error between the original envelope and syntesized curve. The effect of parameter n is illustrated in Figure 4.

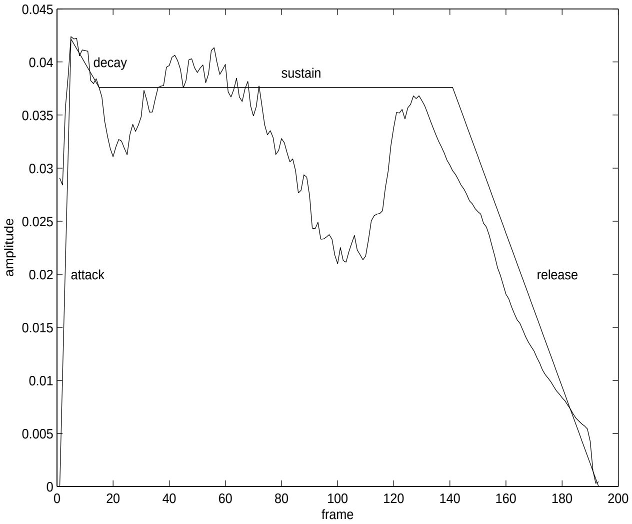

Figure 3: *Original amplitude slope of one harmonic component and the estimated ADSR slope.*

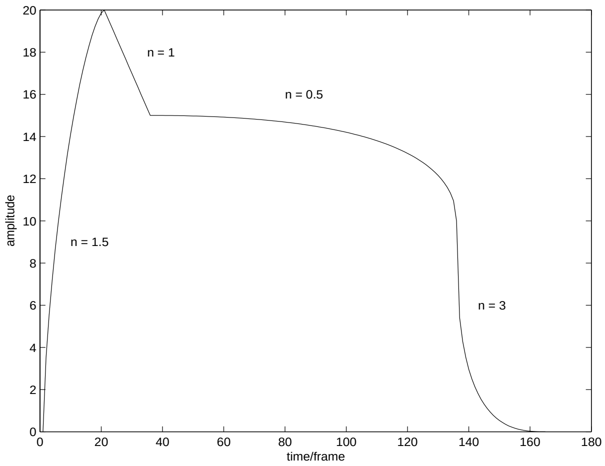

Figure 4: *The effect of parameter n on the shape of an ADSR slope.*

2.3. Combining MFCC and ADSR

The described representations are combined in the following way: First, the MFCCs are calculated in each frame. Second, the temporal evolution of each coefficient is represented using the ADSR. Thus, instead of using the ADSR for the amplitudes the MFCCs are further modeled using the ADSR, producing four time/ MFCC-pairs per each selected MFCC. It is assumed that for a sound harmonic components of which are represented using ADSR, the MFCCs can also be represented using the ADSR. The assumption was validated by listening to the signals that were coded using both representations. An example of MFCCs represented with ADSR is illustrated in Figure 5. There we can see that the rough shape of curve remains but the details are of course lost.

One problem when modeling MFCCs using the ADSR is that unlike the amplitudes of sinusoids, MFCCs are not restricted to non-negative values. In case of negative coefficients, the basic ADSR can not be used. We have solved this by checking whether the curve is positive or negative and in case of negative curve we use the inverse ADSR instead. This is accomplished by simply reflecting the ADSR slope to the other side of x-axis and then finding the similar time/amplitude pairs as before. Otherwise the same curve is modeled.

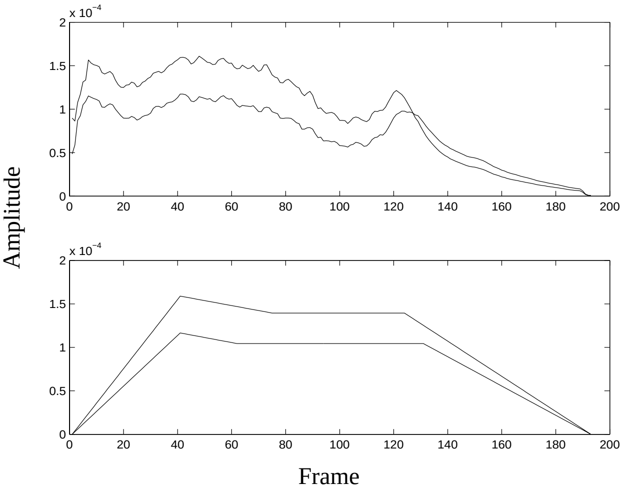

Figure 5: *Two MFCC slopes and the corresponding ADSR slopes.*

Both MFCCs and ADSR alone decrease the amount of data but when used together, the compression rate is remarkably better. The compression rate depends on how long the sound is in the time domain. The longer the sound is the better compression rate is achieved. This is due to the fact that the length of sound does not affect the number of parameters needed to represent it because ADSR uses a fixed number of parameters.

3. APPLICATION TO OBJECT-BASED CODING

The proposed representation for harmonic sounds is used as a part of an object-based coding system. The system has three representations for different components of sounds: harmonic components are modeled using the proposed representation, short transient-like bursts are modeled using a specific transient representation and periodic components which do not fit to the harmonic model are represented as individual sinusoids. The block diagram of the harmonic part of the system is illustrated in Figure 6.

Harmonic sound objects are not restricted to be physical objects which correspond to for example single notes played by one instrument, nor to be perceptual objects which correspond to sound events heard individually. The aim is to find the objects using which a polyphonic music signal can be represented using only a small number of parameters while preserving the perceptual quality of the signal. However, it has been assumed that the objects in this efficient representation correspond closely to physical or perceptual sound events.

At first the frequencies at the harmonic components are esti-

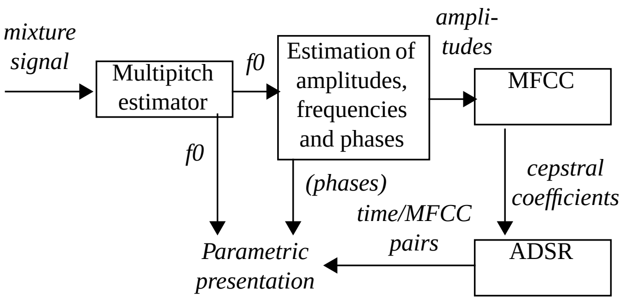

Figure 6: *Block diagram of the harmonic part of the coding system.*

mated using a multipitch estimator (MPE). As a MPE a method which is based on enhanced summary autocorrelation function (ESACF) is used [7]. To find out the frequency of the first harmonics we simply take the highest peaks from the ESACF and assume that these are clean harmonic sounds. There is a threshold above which all the peaks are taken with a restriction that the maximum amount of fundamental frequencies in one frame is fixed.

After the MPE, amplitudes and phases are estimated from the signal spectrum for the estimated frequencies. If fundamental frequencies in consecutive frames are the same or at least close to one another, these sounds are combined to one sound object. If one object is longer than six frames, the MFCC-ADSR method is applied to represent the harmonic components of this object.

The harmonic components are synthesized and subtracted from the original signal to obtain residual, from which transients are estimated. Transients are broadband attack signals which can not be efficiently modeled with sinusoidal modeling. Instead, transform coding is applied to model transient parts of the signal [8]. Finally, sinusoidal analyses are applied to find out the most significant individual sinusoids from the signal.

4. QUALITY EVALUATION

The system is tested to a monophonic signals only. As a quality evaluators we used the Perceptual Audio Quality Measure (PAQM) [9] and informal listening tests. One modification was made to the PAQM algorithm to make it more suitable for comparison of single notes: scaling in three frequency ranges was bypassed, because it gave too good results if the original note had a very little energy in some frequency band and the synthesized note had significant distortion in that band.

In Table 1, the PAQM evaluations for some monophonic signals are presented. As can be seen, the different coding methods does have an effect on PAQM values. Every stage of the processing degrades the quality of sound. ADSR alone degrades the quality just a little but on the other hand the effect is remarkable when MFCCs are used. Furthermore the difference when using MFCCs alone or MFCC+ADSR system is very small.

The informal listening tests also support the idea that sound quality is lower when using MFCC or ADSR but the difference is however surprisingly small. Demonstrations of monophonic signals are available at http://www.cs.tut.fi/~heln/demopage.html.

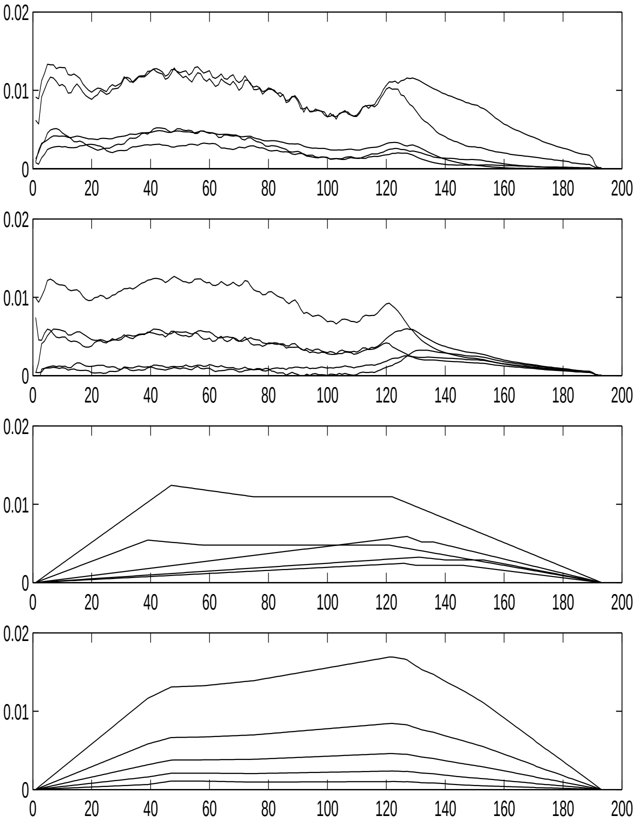

Figure 7: *Curves for the original amplitudes, cepstral coefficients, cepstral coefficients + ADSR and generated amplitudes.*

Also in Figure 7 it can be seen that every stage of sound processing degrades the quality of signal. Especially the details of sound are lost but the rough shape remains.

Table 1: *PAQM values for monophonic signals*

	UNPROCESSED/ log(noise disturbance)	**MFCC/** log(noise disturbance)	**ADSR/** log(noise disturbance)	**MFCC+ ADSR/** log(noise disturbance)
CELLO	-3.2	-2.0	-2.9	-1.9
GUITAR	-5.1	-3.6	-4.5	-3.6
VIOLIN	-1.0	-0.4	-1.1	-0.4
BASS	-3.3	-2.7	-3.3	-3.0

5. CONCLUSIONS

The proposed model aims at efficient representation of harmonic sounds. The rough shape of spectrum is modeled using Mel-cepstral coefficients, and their temporal evolution is further parameterized using attack-decay-sustain-release-model. Algorithm for the estimation of the parameters is explained. The proposed representation is used as a part of an audio-coding system. Qualitative evaluations and informal listening tests show that the model decreases the quality of sounds slightly but preserves perceptually most important features.

6. REFERENCES

[1] X. Serra, “Musical Sound Modeling with Sinusoids plus Noise”, Roads, Pope, Poli (eds.), “*Musical Signal Processing*”, Swets & Zeitlinger Publishers, 1997.

[2] J. D. Markel and A. H. Gray Jr., “Linear Prediction of Speech”, *Springer-Verlag*, New York, 1976.

[3] B. Edler, H. Purnhagen, and C. Ferekidis, "ASAC - Analysis/ Synthesis Audio Codec for Very Low Bit Rates", *100th AES Convention*, Preprint 4179, May 1997.

[4] A. Eronen, “Automatic Musical Instrument Recognition”, *Master of Science Thesis*, Department of Information Technology, Tampere University of Technology, 2001.

[5] S. B. Davis, P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 4, oo. 357-366.

[6] K. Jensen, “Timbre Models of Musical Sounds”, *Ph.D. dissertation*, Department of Computer Science, University of Copenhagen, 1999.

[7] T. Tolonen and M. Karjalainen, “A Computationally Efficient Multipitch Analyses Model”, *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 6, November 2000.

[8] Scott N. Levine, “Audio Representations for Data Compression and Compressed Domain Processing”, *Ph.D. thesis*, Stanford University, 1999.

[9] J. G. Beerends and J. A. Stemerdink: “A perceptual audio quality measure based on a psychoacoustic sound representation” *J. Audio Eng. Soc*., 40:963-978, December 1992.