METAMORPH: REAL-TIME HIGH-LEVEL SOUND TRANSFORMATIONS BASED ON A SINUSOIDS PLUS NOISE PLUS TRANSIENTS MODEL

John Glover, Victor Lazzarini, Joseph Timoney

The Sound and Digital Music Research Group
National University of Ireland, Maynooth
Ireland
john.c.glover@nuim.ie
victor.lazzarini@nuim.ie
jtimoney@cs.nuim.ie

ABSTRACT

Spectral models provide ways to manipulate musical audio signals that can be both powerful and intuitive, but high-level control is often required in order to provide flexible real-time control over the potentially large parameter set. This paper introduces Metamorph, a new open source library for high-level sound transformation. We describe the real-time sinusoids plus noise plus transients model that is used by Metamorph and explain the opportunities that it provides for sound manipulation.

1. INTRODUCTION

When creating software musical instruments or audio effects, we generally want to have the flexibility to create and manipulate a wide variety of musical sounds while providing an intuitive means of controlling the resulting timbres. Ideally we would like to control these instruments and effects in real-time, as this provides valuable feedback for composers and sound designers as well as enabling the possibility of live performance. As it has been shown that the perception of timbre is largely dependent on the temporal evolution of the sound spectrum [1], it seems natural to use a sound model that is based on the frequency spectrum as a tool to manipulate timbre. Spectral models, which represent audio signals as a sum of sine waves with different frequencies and amplitudes, are therefore an excellent choice of tool for performing musical sound transformations. This frequency domain representation of sound is somewhat similar to the analysis performed by the human hearing system, which enables spectral models to provide ways of transforming audio signals that can be perceptually and musically intuitive. However, they generally suffer from the problem of having too many control parameters to allow for meaningful real-time interaction. One solution to this problem is to provide high-level controls that may change many of the spectral model parameters at once but still allow for precise manipulation of the synthesised sound.

This paper introduces Metamorph, a new open source software library for the high-level transformation of sound using a real-time sinusoids plus noise plus transients model. Sinusoidal models in general are described in Section 2, with our sinusoids plus noise plus transients model described in Section 2.2. The high-level transformations that are available in Metamorph are then described in Section 3, with conclusions in Section 4.

2. SINUSOIDAL MODELS

Sinusoidal modelling is based on Fourier's theorem, which states that any periodic waveform can be modelled as the sum of sinusoids at various amplitudes and harmonic frequencies. For stationary pseudo-periodic sounds, these amplitudes and frequencies evolve slowly with time. They can be used as parameters to control pseudo-sinusoidal oscillators, commonly referred to as partials. To obtain the sinusoidal parameters, the Short-Time Fourier Transform (STFT) is often used to analyse an audio stream or a recorded sound file. This results in a list of bin number, amplitude and phase parameters for each frame of analyzed audio, which can then be used to estimate the spectral peak frequency and amplitude values [2]. Although this approach works well for certain musical sounds, problems can arise due to the fact that the entire audio signal is represented using sinusoids, even if it includes noise-like elements (such as the key noise of a piano or the breath noise in a flute note). If any transformation is applied to the model parameters, these noisy components are modified along with the harmonic content which often produces audible artifacts. A sinusoidal representation of noise is also unintuitive and does not provide an obvious way to manipulate the sound in a meaningful manner. These issues lead to the development of sinusoids plus noise models of sound.

2.1. Sinusoids Plus Noise Models

With Spectral Modelling Synthesis (SMS) [3] Serra and Smith addressed the problems that can occur when noisy sounds are modelled as a sum of sinusoids by splitting the audio signal into two components: a deterministic (harmonic) component and a stochastic (residual) component. Equation 1 shows how the output signal s is constructed from the sum of these two components, where $A_p(t)$ and $\theta_p(t)$ are the instantaneous amplitude and phase of the p-th component, and $e(t)$ is the stochastic component at time t.

$$s(t) = \sum_{p=1}^{N_p} A_p(t) cos(\theta_p(t)) + e(t) \quad (1)$$

As it is assumed that the sinusoidal partials will change slowly over time, the instantaneous phase is taken to be the integral of the instantaneous frequency. This is given by Equation 2, where $\omega(t)$

is the frequency in radians per second and p is the partial number.

$$\theta_p(t) = \int_0^t \omega_p(t)dt + \theta_p(0) \quad (2)$$

The sinusoidal parameters are found by computing the STFT, locating sinusoidal peaks in the magnitude spectrum then matching peaks across consecutive frames to form partials. The harmonic component is then generated from the partials using additive synthesis and subtracted from the original signal leaving a noise-like residual signal. This residual component can then optionally be modelled as filtered white noise. The two signal components can now be manipulated independently and then recombined to create the final synthesised sound.

Other sinusoids plus noise models have been proposed since the release of SMS, such as the model introduced by Fitz and Haken in [4]. It also uses the harmonic partials versus noise distinction, but instead uses bandwidth-enhanced oscillators to create a homogeneous additive sound model. The combination of harmonic and stochastic signals is particularly effective when working with sustained notes that have a relatively constant noise component, but in order to successfully manage shorter noise-like signal regions, sinusoidal models have been further extended to include ways to model transients.

2.2. Sinusoids Plus Noise Plus Transients Models

It has been widely recognised that the initial attack of a note plays a vital role in our perception of timbre [5]. These transient signal regions are very short in duration and can often contain rapid fluctuations in spectral content. Although it is possible to model this sort of signal with sinusoids, doing so can result in the same problems that are encountered when trying to model any noisy signal with sinusoids; it is inherently inefficient and does not offer possibilities for meaningful transformations. Transients are also not well modelled as filtered white noise (as is the case with the SMS stochastic component), as they tend to lose the sharpness in their attack and sound dull [6].

Several systems have been proposed that integrate transients with a sinusoids plus noise model. The method described by Masri in [7] aims to reproduce the sharpness of the original transient during synthesis. First a pre-analysis scan of the audio signal is performed in order to detect transient regions, which are defined as being the areas between a note onset and the point at which the onset detection function (based on an amplitude envelope follower) falls below a fixed threshold or reaches a maximum duration, whichever is shorter. This information is then used during sinusoidal analysis to make sure that the edges of the analysis windows are snapped to the region boundaries. During synthesis, the missing overlap at the region boundaries is reconstructed by extrapolating the waveforms from the centres of both regions slightly then performing a short cross-fade. However, this method can not run in real-time due to the need for a pre-analysis scan of the audio signal.

Levine [8] introduced a sinusoids plus noise model that includes transform-coded transients. Note onsets are located using a combination of an amplitude rising edge detector and by analysing the energy in the stochastic component. Transient regions are then taken to be fixed-duration (66 ms) sections immediately following a note onset. The transients are translated in time during time scaling and pitch transposition, however as the primary application of this work was for use in data compression there is no ability to musically manipulate the transients.

Verma and Meng proposed a system that extends SMS with a model for transients in [6]. They show that transient signals in the time domain can be mapped onto sinusoidal signals in a frequency domain using the discrete cosine transform (DCT). However, it is not suitable for real-time applications as it requires a DCT frame size that makes the transients appear as a small entity, with a frame duration of about 1 second recommended. This is far too much a latency to allow it to be used in a performance context.

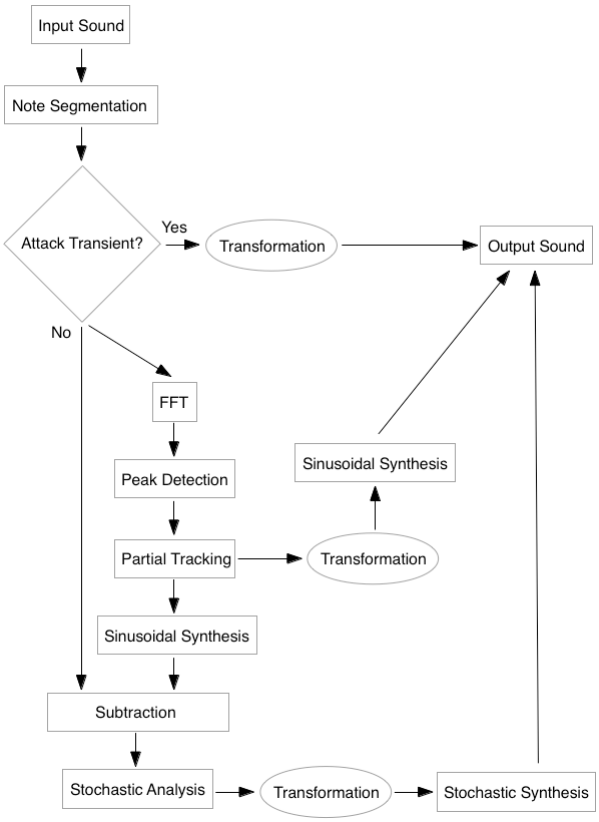

Figure 1: The sinusoids plus noise plus transients model that is used in Metamorph.

Metamorph uses a flexible real-time sinusoids plus noise plus transients model which is summarised in Figure 1. Deterministic and stochastic components are identified using Simpl [9], which provides a choice of several different sinusoidal modelling implementations. Onsets are located using the peak amplitude difference method, one of the best-performing methods discussed in [10]. As this effectively measures errors in the partial tracking stage of the sinusoidal modelling process, it can also be used to provide an indication of the stability of the detected sinusoidal partials in the audio signal. Peaks in the onset detection function should occur at regions where the spectral components in the signal are most unstable or are changing unpredictably. We use this information to locate transient regions, which are defined as the region from the onset until the next local minima in the onset detection function. We also mark the end of the transient region if the root mean square amplitude envelope reaches a local maxima or if the transient reaches a maximum duration of 200 ms (whichever is shorter). An example consisting of an audio signal (a saxophone

sample), our onset detection function and the corresponding transient region is shown in Figure 2. More information on this process is given in [11]. When no transformations are applied to the signal, transients in Metamorph are simply blocks of unmodified raw sample values, with no sinusoidal analysis or synthesis performed during transient regions. During synthesis, we extend the output of the transient region slightly (currently for a single 512 sample frame) in order to perform a short cross-fade between the unmodified sample values in the transient region and the synthesised signal in the note region that follows. The sample values in transient regions may be altered if a transformation is applied however, as we discuss in Section 3.

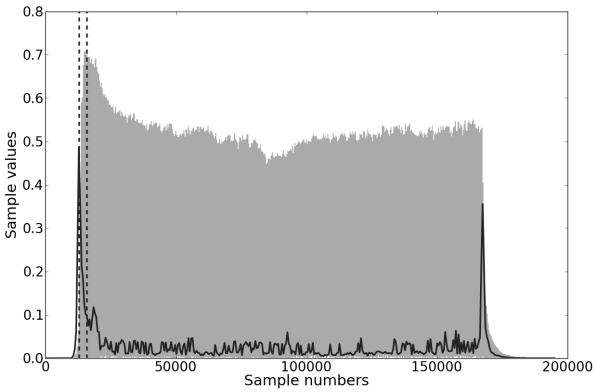

Figure 2: Saxophone sample (grey), onset detection function (solid black line) and detected transient region (between vertical black dashed lines).

3. METAMORPH: HIGH-LEVEL SOUND TRANSFORMATIONS

Metamorph is a new open source library for performing high-level sound transformations based on a sinusoids plus noise plus transients model. It is written in C++, can be built as both a Python extension module and a Csound opcode, and currently runs on Mac OS X and Linux. It is designed to work primarily on monophonic, quasi-harmonic sound sources and can be used in a non-real-time context to process pre-recorded sound files or can operate in a real-time (streaming) mode. Here we define a real-time analysis/synthesis system to be one that can operate with low enough latency and computational requirements so that it is usable in a live musical performance context. While there is no absolute rule that specifies how much latency is acceptable in a live performance context, Metamorph's default latency of 512 samples (or about 11.6 ms) should meet these requirements in many cases. The computational requirements for Metamorph vary depending on the supplied sinusoidal modelling parameters, the type of transformation being applied and on the nature of the sound source itself. However as an example, the noisiness and transience transformation (described in Section 3.2) streaming Csound opcode with default parameters requires approximately 20% of one CPU core on a 2.4 GHz Mac Intel Core 2 Duo processor.

Metamorph is available under the terms of the GNU General Public License (GPL). For more information and to get the software, go to `http://www.johnglover.net`. In this section, we describe the sound transformations that are currently available in Metamorph.

3.1. Harmonic Distortion

As described in [12], the harmonic distortion of a sound is a measure of the degree of the deviation of the measured partials from ideal harmonic partials. The Metamorph harmonic distortion transformation allows the user to alter the deviation of each synthesised partial in a sound from the ideal harmonic spectrum according to Equation 3, where i is the partial number, f is the analysed partial frequency, F_0 is the estimated fundamental frequency, α is the input parameter (between 0 and 1) and F is the output frequency of the synthesised partial.

$$F_i = (\alpha \times f_i) + ((1 - \alpha) \times (F_0 \times i)) \quad (3)$$

3.2. Noisiness and Transience

The noisiness [12] of a synthesised frame is calculated by taking the ratio of the amplitude of the residual component to the total signal amplitude. Metamorph allows the user to easily adjust this balance by altering the amplitudes of the deterministic and stochastic components independently. It also enables the independent manipulation of the amplitude of transient regions. We call this effect changing the signal *transience*.

3.3. Spectral Envelope Manipulation

A spectral envelope is a curve in the spectrum of an audio signal that approximates the distribution of the signal's energy over frequency. Ideally this curve should pass through all of the prominent peaks of the frame and be relatively smooth, preserving the basic formant structure of the frame without oscillating too much or containing discontinuities. Spectral envelopes in Metamorph are calculated using the discrete cepstrum envelope (DCE) method [13] which provides a smooth interpolation between the detected sinusoidal peaks. However, further comparison with the true envelope method [14] is desirable in future, as it seems to produce spectral envelopes that are as good (if not better) than the DCE and it can now be computed efficiently [15]. This was not an immediate priority as the main problem that the authors in [15] had with the DCE was that it required a potentially computationally expensive fundamental frequency analysis or other means of identifying spectral peaks, but this is already a requirement for other parts of the sinusoidal modelling process so there is no extra cost associated with this step in Metamorph.
The spectral envelope transformation in Metamorph allows the user to alter the amplitudes of the synthesised sinusoidal partials to match the corresponding amplitude values in a different spectral envelope. This can be a fixed envelope, or the user can specify a different sound source to use as the target envelope. The user may also alter the synthesised partial amplitudes based on a linear interpolation between the original spectral envelope and the target envelope. Although similar techniques can be performed using the Phase Vocoder [15], spectral envelope manipulation using sinusoidal models enables the preservation (or independent manipulation) of the stochastic signal component, offering greater possibilities for sound transformation. In Metamorph this is taken a step further, enabling the alteration of a spectral envelope while preserving the initial note attack.

3.4. Transposition

Sounds can be transposed in Metamorph in two different ways. Both techniques involve initially multiplying the frequency values of all synthesised partials by the same factor. The second process additionally adjusts all of the partial amplitude values so that they match those of the original spectral envelope. The latter approach preserves the original formant structure, which results in a more natural sounding transposition for certain types of sounds. The transient region is left unmodified for both types of transposition.

3.5. Time Scaling

Time scaling is the only Metamorph transformation that is not available in real-time mode. The time scaling algorithm works by keeping the analysis and synthesis frame rates identical, but instead of passing each analysis frame directly to the synthesis module, frames may be repeated (or skipped) depending on the required time scale factor. This approach does not result in any synthesis artifacts or discontinuities as the synthesis module interpolates smoothly between input frames, and has been shown to produce high-quality time scaling [16]. Frames from transient regions are treated differently however; they are always passed to the synthesis module in the original order, with the sample values passed unmodified to the output so that the transient is maintained. This means that the time scale factor has to be adjusted slightly during non-transient regions in order to make sure that the final output signal is of the required length.

3.6. Transient Processing

Most transformations in Metamorph aim to preserve the original transient region, but it is also possible to explicitly alter the output transient. The most basic effect is to either filter the transient using either low- or high-pass filters, which although relatively simple can have quite a large impact on the resulting sound. The transient can also be removed altogether. Another interesting effect is transient substitution, where the transient regions in the audio signal can be replaced by a different set of audio samples (which may or may not themselves be transients). This allows for the creation of various hybrid instruments, for example combining the attack of a drum sound with a sustained woodwind instrument tone.

4. CONCLUSIONS

This paper introduced Metamorph, a software library which provides a new environment for performing high-level sound manipulation. It is based on a real-time sinusoids plus noise plus transients model, which enables it to perform a collection of flexible and powerful sound transformations. Metamorph is free software, can be used as a C++ library, Python extension module or set of Csound opcodes and is available under the terms of the GNU General Public License. To download it or for more information go to `http://www.johnglover.net`.

5. REFERENCES

[1] John M. Hajda, *Analysis, Synthesis, and Perception of Musical Sounds*, chapter The Effect of Dynamic Acoustical Features on Musical timbre, pp. 250–271, Springer, 2007.

[2] Robert McAulay and Thomas Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-34, no. 4, August 1986.

[3] Xavier Serra and Julius O. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decompostion," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, Winter 1990.

[4] Kelly Fitz, *The Reassigned Bandwidth-Enhanced Method of Additive Synthesis*, Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA, 1999.

[5] John M. Grey, *An Exploration of Musical Timbre*, Ph.D. thesis, Stanford University, USA, 1975.

[6] Tony S. Verma and Teresa H. Y. Meng, "Extending spectral modeling synthesis with transient modeling synthesis," *Computer Music Journal*, vol. 24, no. 2, pp. 47–59, Summer 2000.

[7] Paul Masri, *Computer Modeling of Sound for Transformation and Synthesis of Musical Signals*, Ph.D. thesis, University of Bristol, United Kingdom, 1996.

[8] Scott Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, Ph.D. thesis, Stanford University, 1998.

[9] John Glover, Victor Lazzarini, and Joseph Timoney, "Simpl: A Python library for sinusoidal modelling," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, September 2009.

[10] John Glover, Victor Lazzarini, and Joseph Timoney, "Real-time detection of musical onsets with linear prediction and sinusoidal modeling," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 68, 2011.

[11] John Glover, Victor Lazzarini, and Joseph Timoney, "Real-time segmentation of the temporal evolution of musical sounds," in *The Acoustics 2012 Hong Kong Conference*, Hong Kong, China, May 2012.

[12] Xavier Serra and Jordi Bonada, "Sound transformations based on the sms high level attributes," in *Proceedings of the International Conference on Digital Audio Effects*, Barcelona, Spain, 1998.

[13] Thierry Galas and Xavier Rodet, "An improved cepstral method for deconvolution of source filter systems with discrete spectra: Application to musical sound signals," in *Proceedings of the International Computer Music Conference (ICMC'90)*, Glasgow, Scotland, 1990, pp. 82–84.

[14] S. Imai and Y. Abe, "Spectral envelope extraction by improved cepstral method," *Electron. and Commun. in Japan*, vol. 62-A, no. 4, pp. 10–17, 1979.

[15] Axel Röbel and Xavier Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx-05)*, Madrid, Spain, September 2005.

[16] J Bonada, "Automatic technique in frequency domain for near-lossless time-scale modification of audio," in *Proceedings of the International Computer Music Conference (ICMC'00)*, Berlin, Germany, 2000, pp. 396–399.