MULTIRESOLUTION SINUSOIDAL MODELING FOR WIDEBAND AUDIO WITH MODIFICATIONS

Scott N. Levine, Tony S. Verma, Julius O. Smith III

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-8180

ABSTRACT

In this paper, we describe an computationally efficient method of generating more accurate sinusoidal parameters {*amplitude, frequency, phase*} from a wideband polyphonic audio source in a multiresolution, non-aliased fashion. This significantly improves upon previous work of sinusoidal modeling that assumes a single-pitched monophonic source, such as speech or an individual musical instrument, while using approximately the same number of sinusoids. In addition to a more general analysis, we can now perform high-quality modifications such as time-stretching and pitch-shifting on polyphonic audio with ease.

1. INTRODUCTION

Sinusoidal modeling has been developed as flexible, parametric method of representing speech [10] and musical instruments [8]. These methods assume that most speech and audio signals can be well represented by many time-varying sinusoids.[1] The problem is to model the audio using time-varying sinusoids in an efficient and perceptually meaningful manner.

We solve this problem of parameter estimation by first splitting the input signal into octave-spaced, oversampled, non-aliased subbands. Then, we perform sinusoidal modeling individually on each subband signal. Not only is this method efficient in the number of sinusoids needed to faithfully represent the signal, but it is efficient in overall complexity and sounds much better than using a single-subband sinusoidal model.

2. WINDOW LENGTH TRADE-OFFS

A challenging problem for polyphonic sinusoidal analysis is wisely choosing the window length. Because of the near logarithmic scale of pitch perception, we need very long windows in order to accurately estimate the pitch of low frequency partials. The higher the frequency of the partial, the less frequency resolution the analysis needs. The tradeoff in improved frequency resolution is of course, worse time resolution.

Over the period of a single analysis frame, the algorithm estimates the amplitude, frequency, and phase of any sinusoids it believes to be present. The time resolution of these solutions is only as fine as the window length, itself. Thus, we desire as short of a window as possible. The traditional method of synthesizing a new onset partial in sinusoidal modeling is to ramp up the amplitude from zero in the middle of the previous frame F_{i-1} to A_0 in the middle of the current frame F_i. This is illustrated in Figure 1. Thus, the shorter the window, the shorter the ramp-up duration; in addition, the sinusoid onset will better localized in time.

This ramp-up effect is analogous to the pre-echo problem in audio data compression. The onset of the synthesized sinusoid occurs in the middle of the previous frame (before the original sinusoid begins) just as the quantization noise in transform coding occurs at the beginning of the current frame (before the onset of the original waveform).

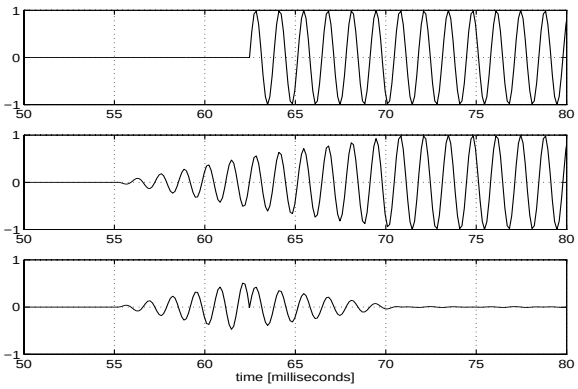

Figure 1: The top figure is the original sinusoid. The middle figure is the synthesized sinusoid, with the amplitude ramped from the previous frame. The bottom figure is the error difference signal between the original and the synthesized signal.

Previous works [13, 11] use a pitch-synchronous analysis to determine the window length. As often as once a frame, a pitch estimate is calculated. Then, the analysis window is adjusted accordingly; the lower the pitch, the longer the window. The window is thus guaranteed to correctly estimate the fundamental frequency, and all frequencies above it. This pitch-estimation does not solve the pre-echo problem, but does assure that all partials above the fundamental will be resolved. In fact, the lower the fundamental pitch, the longer the window, and the worse the pre-echo problem.

This approach works reasonably well if the input signal can be assumed to be monophonic and single-pitched. For polyphonic audio, it is impractical to attempt to discern multiple pitches. To

[1] Later, it was shown that one can model the residual of sinusoidal modeling as noise [14], or model the residual as transients+noise [7]. In this paper, we will only concentrate on the problem of estimating sinusoids.

solve this problem, we split the input signal into several bandlimited frequency ranges, and then design a window length for each channel individually.

3. MULTIRESOLUTION FILTERBANK

There have been several different previous approaches to solving the sinusoidal parameter estimation problem in a multiresolution manner. One method is to input a signal through a octave-spaced, critically sampled, wavelet filter bank [6, 1, 12], and perform sinusoidal modeling on the channels. This results in relatively low complexity, but there is no known way to eliminate all aliasing between channels in the filter bank. Therefore, each channel estimates sinusoidal parameters of the actual bandpassed-octave signal, in addition to parameters of the aliased octaves adjacent in frequency. It is possible to reduce these cross-talk aliasing terms [1, 4, 15], but complexity is now raised, and the results have not been sufficient for high quality, wideband sinusoidal modeling. For other alternative methods, refer to the discussion in [9].

In this paper, we use an octave-spaced, complementary filter bank [5] as the front end to a bank of sinusoidal modeling algorithms. Each channel output goes into a separate sinusoidal modeling block, with its own window and analysis parameters, as seen in Figure 2. Notice that there is no synthesis filter bank. The $\{A, \omega, \phi\}$ parameters are extracted from the several sinusoidal modeling blocks, and then are fed into a sinusoidal synthesizer.

We thus avoid the two main problems of previous schemes: with the filter bank discussed in the next section, we avoid the aliasing cross-talk problem as seen in wavelet filter bank front-ends. By introducing downsampling into the filter bank, we avoid the high costs of storage, memory, and complexity as seen in the constant-Q non-decimated filter banks, or the multiple FFT schemes.

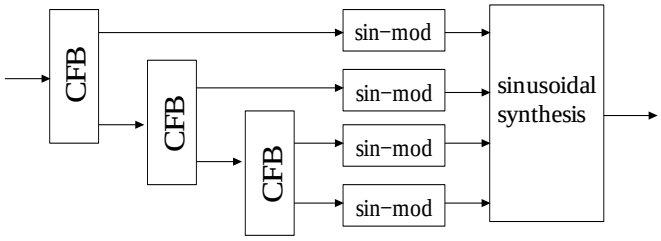

Figure 2: The input gets split into highpass and lowpass signals three times by the *CFB* blocks, or the complementary filter banks, as shown in Figure 3. The four octaves of output signals each get fed into the *sin-mod* blocks, or the sinusoidal modeling blocks. It is here that the sinusoidal parameters $\{A_k^i, \omega_k^i, \phi_k^i\}$ are estimated for the k^{th} partial of the i^{th} octave. The *sinusoidal synthesizer* can be implemented either as a bank of oscillators or using IFFTs.

4. ALIAS-FREE SUBBANDS

The filter bank is also designed to assure that the subband signals are alias-free. There is overlap in frequency ranges between the channels, but no frequencies get folded over due to the subsampling. This filter bank structure is based upon the Laplacian pyramid structure [3] from the multiresolution image compression world. The enhancement made to the Laplacian structure is the intermediate filter H_b [5], as seen in Figure 3. The filter H_b eliminates the spectral copy shifted to $\omega = \pi$ after the lowpass filter H_d and downsampling, If H_d were an ideal lowpass (brickwall) filter, then there would be no overlapping spectral copy; but this is not the case in practice[2].

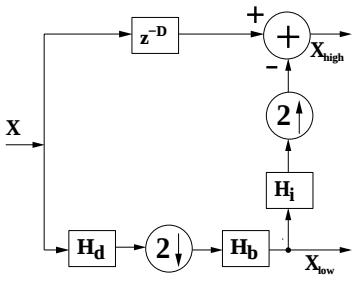

Figure 3: The two channel, complementary filter bank. H_d and H_i are the decimation and interpolation filters, respectively. Each are designed as FIR lowpass filters, with a cutoff frequency near $\pi/2$.

It is important to guarantee no aliasing in the subbands; if there were aliasing, the sinusoidal modeling blocks may generate parameters due to the aliased sinusoids in addition to the actual sinusoids. Since there is no synthesis filter bank, we cannot rely upon aliasing cancellation in synthesis. The sinusoidal synthesizer generates sinusoids from either a bank of oscillators, or from a block IFFT.

For this benefit of alias-free subbands, the filter bank can no longer critically sampled; but rather, it is oversampled by a factor of two. This factor of two is independent upon the number of octaves in the system. This is contrast to the methods of [1, 6], whose complexity and data rate grew linearly as a function of the number of octaves.

To examine the subband signals to be approximated, examine figure 5 for the magnitude spectrum of the four output channels of the octave-based complementary filter bank. Figure 4 shows the magnitude spectrum of the original saxophone note. Notice that the lower plots in Figure 5 have their harmonic partials stretched further and further. This is due to the progressive downsampling present in each channel.

Notice also that there is no audible aliasing between channels. There are some partials that exist in both channels (for example, a partial at $1150Hz$ occurs in both of the lowest two channels). But, when synthesized with the correct phase, the two partials constructively sum to the single, original partial. Also notice the frequency regions of nearly zero energy in the plots of Figure 5. This is also due to the fact that the filter bank is two times oversampled. Thus, there is almost half of the bandwidth with zero energy. If the filter bank were critically sampled, then the plots would have no 'dead-zones', or frequency regions lacking energy. But, the energy would be partially aliased.

5. MULTIRESOLUTION WINDOWING

Now that we have introduced the multiresolution scheme, as pictured in Figure 2, we will examine the time-frequency tiling of this approach, along with its pre-echo characteristics.

[2] The operators in figure 3 could be commuted into one lowpass filter, and a single downsampler for a lower complexity filterbank.

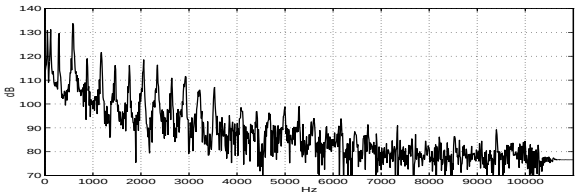

Figure 4: The magnitude spectrum of the original saxophone note, sampled at 22 kHz.

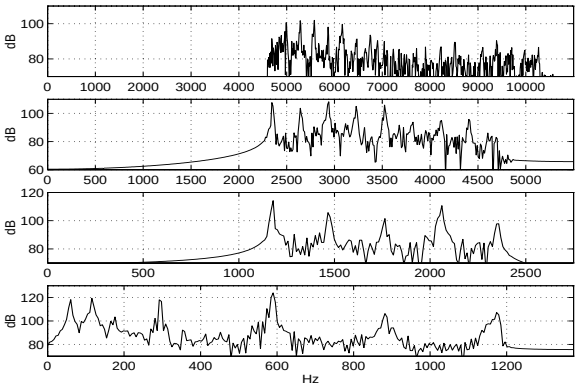

Figure 5: The magnitude spectra of the four channels of the filter bank, with the input shown in Figure 4. The top plot is the highest frequency octave band (not downsampled) and the bottom plot is the lowest frequency octave (downsampled by 8).

Each of the *sin-mod* diagrams in Figure 2 first windows its incoming data. All of the windows are of the same length in each of the *sin-mod* blocks. But, the data rate in each of the octaves is different. If the original data rate is f_s, then the data rate in the top octave is f_s, the data rate in the octave directly below is $f_s/2$, then $f_s/4$, and the data rate in the lowest octave is $f_s/8$. So, the data rates are different and the window lengths stay the same. But, one can also interpret this in the following way: the effective data rates are the same, and the effective window lengths are different. In this manner, the highest octave window would be of length L samples, the window in the octave below would be of an effective length of $2L$, then $4L$, and finally the lowest octave would have a window length of $8L$.

This effect can be seen in Figure 6. Consider the top octave, initially. Its effective window length is L, and is updated every L points[3] . Once the windowing has taken place, we estimate the sinusoidal parameters, using approximate maximum likelihood estimate techniques based on [7] from sinusoidal modeling, and originally developed by [16]. We then hop another window length (again, 1:1 hop size only for simplicity of discussion), and repeat the parameter estimation. Once a sufficient number of parameters from adjacent windows are recorded, we then begin the process of sinusoidal peak continuation, as developed by [14].

This windowing and parameter estimation is also performed for the other three lower octaves. But, the parameter update rate is halved for every octave lower than the top one. For example, in the second highest octave, sinusoidal parameters are computed

[3] In practice, we use an overlap of 2:1 or 4:1 for windowing. In Figure 6 and in this discussion, we discuss an overlap of 1:1 for clarity.

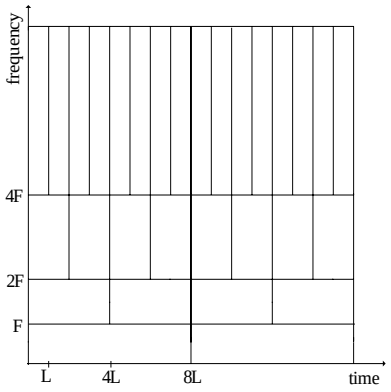

Figure 6: The tiling of the time-frequency plane.

at times $\{2L, 4L, 6L, 8L, ...\}$, while the parameters at the lowest octave are computed at times $\{8L, 16L, 24L, ..., \}$. Therefore, the higher the octave, the denser the time sampling of sinusoidal parameters.

6. MULTIRESOLUTION PRE-ECHO

With the new system implemented, we have now limited the extent of pre-echo by frequency region. Because the effective window length is shorter at the higher octaves, the pre-echo is correspondingly limited to the length of this shorter window. At the lowest octave, we still need long effective windows for proper frequency resolution, and there still will be some amount of audible pre-echo.

After extensive listening tests, it seems that what matters in pre-echo is not how many milliseconds of pre-echo is present, but rather how many *periods* of a signal exists as pre-echo at each partial frequency. Therefore, at high frequencies, there may be 1-3 milliseconds of pre-echo, but several periods are still contained. At low frequencies, there may be 10-20 milliseconds of pre-echo, but the amount of periods in this regions is relatively similar to that of the high frequencies.

7. RESIDUAL MODELING

The multiresolution sinusoidal modeling approach stated in this paper is actually the first of a multi-step process to handle complex audio signals. Once all the valid sinusoids are found, they can be synthesized, subtracted from the original (delayed) signal, and then produce a residual. This residual, which is composed of mostly transients and noise, is then processed through our transient modeling synthesis [TMS] [17] algorithm. The TMS algorithm can parametrically model transients with very fine time resolution.

The residual from the transient modeling synthesis can then be modeled by using a frame-based, time-varying filtered noise algorithm with some sort of excitation, in the same manner as speech compression algorithms. Alternatively, a psychoacoustic, transform-based algorithm [2] could efficiently allocate bits to the signal energy of this second residual not masked by the sinusdoidal modeled synthesis or the transient modeled synthesis.

8. MODIFICATIONS

As shown by [14], it is relatively straightforward to to time-scale and pitch-scale modify audio when it is represented in tracks of sinusoids. Both time-scale and pitch-scale modifications can be made independently.

The ratio of synthesis window length to analysis window length in each octave-spaced channel is equal to the time-scaling factor. The amplitude, frequency, and phase parameters in their respective sinusoidal tracks are now simply interpolated over a different hop-size length. To pitch-scale modify audio, the frequency parameters are all scaled accordingly (and eliminated if scaled above $fs/2$). Future versions could include alogorithms for preserving the spectral magnitude envelope; while this is necessary for speech and some musical instruments, it is not clear if this is useful for polyphonic audio.

9. RESULTS

After listening to many genres of $22kHz$-sampled music tested with both one band and four bands of sinusoidal modeling, it seems that drums gave one band sinusoidal modeling the most problems. Any cymbal or snare hits generate short-time, broadband energy that can not be well represented by one-band, $40ms.$ windowed sinusoids. This results in a distorted sound, that blurs or eliminates any sharp attacks. In this current multiresolution representation, the shortest windows are about $5ms.$ long in the top octave ($5 - 11$ kHz), which seems about short enough for most attacks. These results are not perceptually lossless by using sinusoids only, but are much better than using one band.

In addition, the residual is now better time-localized and has less pre-echo errors in the higher octaves. Therefore, the error signal will be better represented by the residual models described in the section 7.

Not only does the four-band system sound better than single-band sinusoidal modeling, but the number of sinusoids estimated per frame at any given time is approximately the same between systems. For reasonable quality, the system extracts anywhere between $60 - 80$ sinusoids for the single band system, and $10 - 20$ sinusoids per octave-band channel in the four-band system.

10. CONCLUSIONS

We have proposed a new multiresolution sinusoidal parameter estimation algorithm that has the computational efficiency of wavelet filter bank front-ends, but without any of their aliasing problems. In return, we have twice as much intermediate subband data as with a critically-sampled wavelet filter bank. But more importantly, there is no increase in the resulting number of sinusoidal parameters due to the oversampling of the subband data.

With this structure in place, it is now reasonable to perform sinusoidal modeling on polyphonic, wideband audio without having to tweak many parameters or attempt a pitch-esimation in parallel. All modifications previously performed on monophonic sources, such as time-streching and pitch-shifting, can now be easily extended more generally to wideband audio.

11. REFERENCES

[1] D. Anderson. Speech analysis and coding using a multiresolution sinusoidal transform. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Atlanta*, pages 1037–1040, 1996.

[2] M. Bosi, K. Brandenburg, S. Quackenbush, L.Fielder, K. Akagiri, H.Fuchs, M.Dietz, J.Herre, G.Davidson, and Y.Oikawa. ISO-IEC MPEG-2 Advanced Audio Coding. *Audio Engineering Society Convention*, 1996.

[3] P.J. Burt and E.A. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4), April 1983.

[4] B. Edler. Aliasing reduction in sub-bands of cascaded filter banks with decimation. *Electronic Letters*, 28(12):1104–1106, 1992.

[5] N.J. Fliege and U. Zolzer. Multi-complementary filter bank. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Minneapolis*, 1993.

[6] M. Goodwin and M. Vetterli. Time-frequency signal models for music analysis, transformation, and synthesis. *IEEE SP International Symposium on Time-Frequency and Time-Scale Analysis*, 1996.

[7] A. Hamdy, K. Ali and Tewfik H. Low bit rate high quality audio coding with combined harmonic and wavelet representations. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Atlanta*, 1996.

[8] J. O. Smith III and X. Serra. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. *Proceedings of the 1987 International Computer Music Conference, Champaign-Urbana*, 1987.

[9] S. Levine, T. Verma, and J.O. Smith. Alias-free, multiresolution sinusoidal modeling for polyphonic, wideband audio. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY*, 1997.

[10] R. McAulay and T. Quatieri. Speech transformations based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, Signal Processing*, 34, December 1986.

[11] R. McAulay and T. Quatieri. *Speech Coding*, chapter 4, pages 121–173. Elsevier Science B.V., 1995.

[12] M. Rodriguez-Hernandez and F. Casajus-Quiros. Improving time-scale modification of audio signals using wavelets. *ICSPAT*, 2:1573–1577, 1994.

[13] X. Serra, J. Bonada, P. Herrera, and R. Louriero. Integrating complementary spectral models in the design of a musical synthesizer. *Proceedings of the 1997 International Computer Music Conference, Greece*, 1997.

[14] X. Serra and J. O. Smith III. Spectral modeling synthesis: A sound analysis / synthesis system based upon a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, winter 1990.

[15] B. Tang, A. Shen, G. Pottie, and A. Alwan. Spectral analysis of subband filtered signals. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Detroit*, 1995.

[16] D. Thomson. Spectral estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9), September 1982.

[17] T. Verma, S. Levine, and T. Meng. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. *Proceedings of the 1997 International Computer Music Conference, Greece*, 1997.