Alias-free, Multiresolution Sinusoidal Modeling for Polyphonic, Wideband Audio

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 improved 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. In addition to a more general analysis, we can now perform high-quality transformations 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 [1] and musical instruments [2]. These methods assume that most speech and audio signals can be well represented by many time-varying sinusoids.[1] The trick is estimating these 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, 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 as well.

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 records 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.

[1]Later, it was shown that one can model the residual of sinusoidal modeling as noise [3], or model the residual as transients+noise [4]. In this paper, we will only concentrate on the problem of estimating sinusoids.

This ramp-up effect is analogous to the pre-echo problem in audio data compression. The onset of the synthesized sinusoid occurs before the original sinusoid just as the quantization noise in transform coding occurs before the onset of the input audio.

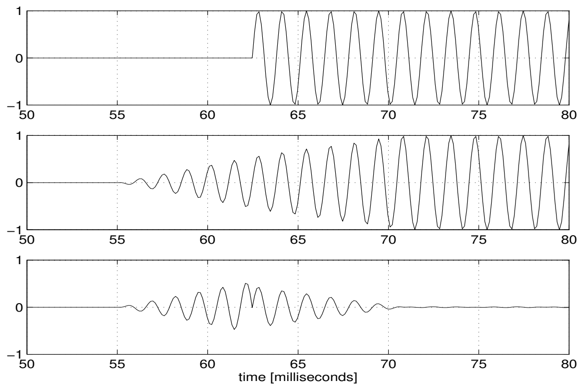

Figure 1: The top figure is the original sinusoid. The middle figure is the synthesized sinusoid, with the amplitude ramped from the previous frame. The bottom figure is the error difference signal between the original and the synthesized signal.

Previous works [5, 6] 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 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 Approach

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 [7, 10, 11], 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 [10, 13, 14], but complexity is now raised, and the results have not been sufficient for high quality, wideband sinusoidal modeling.

Another multiresolution approach to sinusoidal modeling is to use a parallel bank of constant-Q bandpass filters as a front end [7, 12]. Then, one can perform sinusoidal modeling on each bandpass filter output. Although this method works well, there is no downsampling, and the structure is highly oversampled. The amount of data storage and complexity increases linearly with the number of bandpass filters.

A third recent approach is to take multiple sliding FFTs of the same audio input [10]. Each FFT has a different window length, and thus a different time-frequency plane tiling. The long windowed, low frequency FFT output is fed into sinusoidal modeling for low frequency sinusoids; similarly, short windowed, high-frequency FFT data is fed into sinusoidal modeling for high frequency sinusoids. Complexity in this algorithm also linearly increases with the number of octaves, of FFTs sets, taken.

In this paper, we use an octave-spaced, complementary filter bank [15] 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.

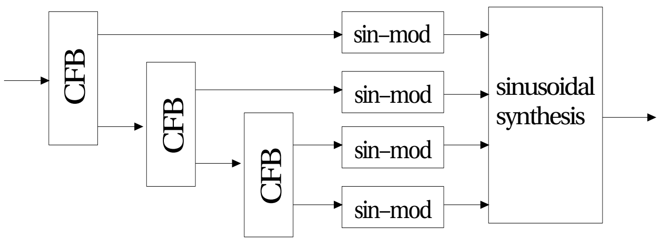

Figure 2: The input gets split into highpass and lowpass signals three times by the *CFB* blocks, or the complementary filter banks, as shown in Figure 3. The four octaves of output signals each get fed into the *sin-mod* blocks, or the sinusoidal modeling blocks. It is here that the sinusoidal parameters $\{A_k, \omega_k, \phi_k\}$ are estimated. 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 [16] from the multiresolution image compression world. The enhancement made to the Laplacian structure is the intermediate filter H_b, 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, as seen in Figure 4. 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.

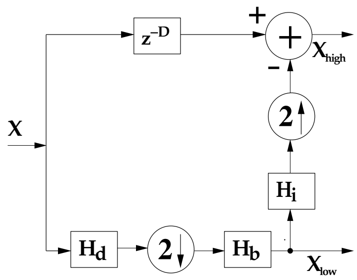

Figure 3: The two channel, complementary filter bank. H_d and H_i are the decimation and interpolation filters, respectively. Each are designed as FIR lowpass filters, with a cutoff frequency near $\pi/2$.

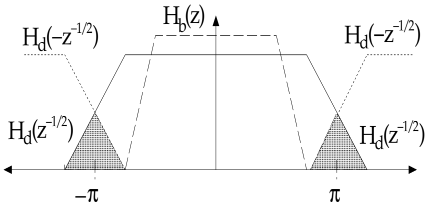

Figure 4: The downsampled filter has aliased energy around $\omega = \pi$ due to the sum of $H_d(z^{-1/2}) + H_d(-z^{-1/2})$. To eliminate this aliased (and shaded) energy, we then apply the intermediate filter, H_b (solid). The result is an output octave channel whose aliased energy has been attenuated below the stopband region of $H_b(z)$.

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 [10, 7], whose complexity and data rate grew linearly as a function of the number of octaves.

In addition to reducing the amount of data, the downsampling process also aids the frequency estimation process at the lower octaves. First, examine Figure 5 to see the magnitude spectrum of a saxophone note. We show in Figure 6 the magnitude spectrum of the four output channels of the octave-based complementary filter bank. Notice that the lower plots in Figure 6 have their harmonic partials stretched further and further. This is due to the progressive downsampling present in each channel. It is simpler for a peak detection algorithm to sucessfully find peaks that are spread apart, rather than those that are closely bunched together.

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 6. 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.

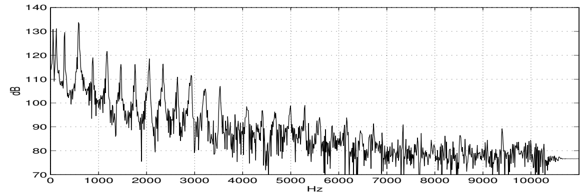

Figure 5: The magnitude spectrum of the original saxophone note, sampled at 22 kHz.

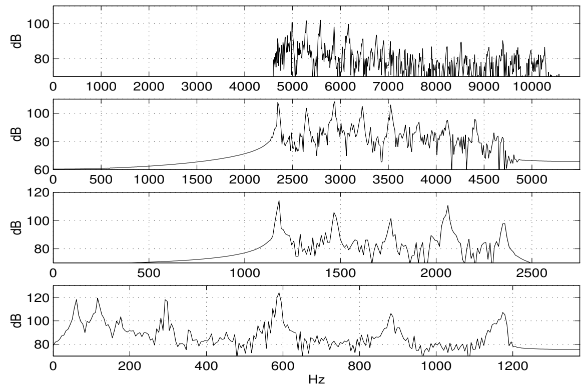

Figure 6: The magnitude spectra of the four channels of the filter bank, with the input shown in Figure 5. The top plot is the highest frequency octave band (not downsampled) and the bottom plot is the lowest frequency octave (downsampled by 8).

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.

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 7. Consider the top octave, initially. Its effective window length is L, and is updated every L points[2] . Once the windowing has taken place, we estimate the sinusoidal parameters $\{A_0, \omega_0, \phi_0\}$, using approximate maximum likelihood estimate techniques based on [4] from sinusoidal modeling, and originally developed by [17]. 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 [3]. This process keeps only "stable" partials that last for more than a given length of time, and deletes spurious sinusoids.

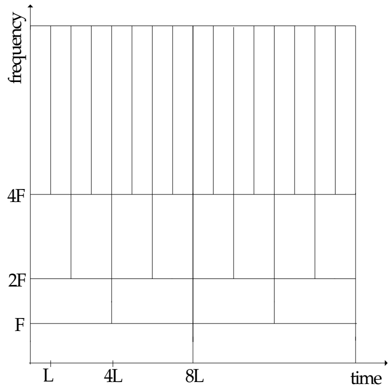

Figure 7: The tiling of the time-frequency plane.

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 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.

To illustrate this point, examine Figure 8. There are four plots, each showing sinusoidal trajectories over time in their particular octave. Notice how the time resolution doubles for every higher octave. The original recording is 600 milliseconds of a jazz trio comprised of drums, bass, and saxophone (sampled at 16 kHz).

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

[2] In practice, we use an overlap of 2:1 or 4:1 for windowing. In Figure 7 and in this discussion, we discuss an overlap of 1:1 for clarity.

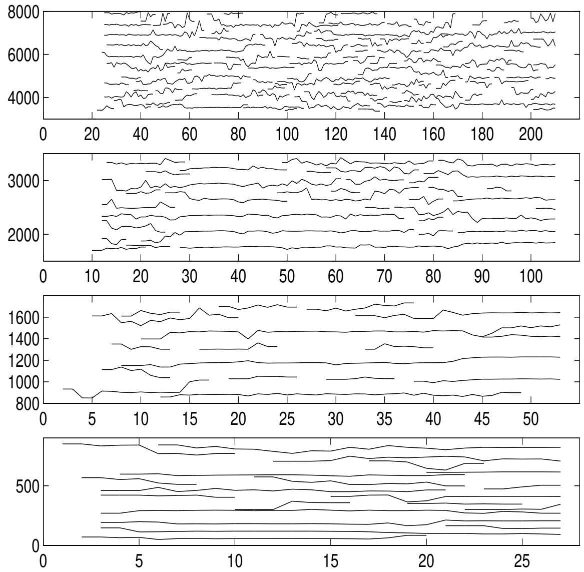

Figure 8: The sinusoidal trajectories from four octave-spaced channels. Each of the horizontal axis is measured in sinusoidal update timings. Even though each octave has a different number of updates, the absolute time scale [in milliseconds] is still the same across octaves. The vertical axis on each subplot is linear within its octave of frequency [Hz].

of periods in this regions is relatively similar to that of the high frequencies.

Other people have devised window switching methods to solve this pre-echo problem. The audio compression standard, MPEG-2 AAC [9], switches between single long windows of length 2048 samples to series of short windows of length 256 samples. The window length is switched according to their *perceptual entropy*, which usually increases during transient regions. By maintaining short windows during transients, the algorithm time-limits the quantization noise within a short window. This reduces the pre-echo audibility. Another approach is to use dynamic programming to find an optimal time segmentation [8], which is a more general approach to the pre-echo problem, but also more complex.

The multiresolution sinusoidal modeling approach stated in this paper is actually the first of a two step process to handle transients and pre-echo. 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] [18] algorithm. The TMS algorithm can parametrically model transients with very fine time resolution.

7. Conclusions

We have proposed a new multiresolution sinusoidal parameter estimation algorithm that has the complexity 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.

References

1. McAulay, M. and Quatieri, T, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, August 1986.
2. Smith, J.O., and Serra, X., "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation", *International Computer Music Conference*, 1987.
3. Serra, X., and Smith, J. O., "Spectral modeling synthesis: A sound analysis / synthesis system based upon a deterministic plus stochastic decomposition", *Computer Music Journal*, 14(4):12-24, Winter 1990.
4. Hamdy, K.N., Ali, A., and Tewfik,A.H., "Low bit rate high quality audio coding with combined harmonic and wavelet representations", *ICASSP-1996*.
5. Serra, X, "Musical sound modeling with sinusoids plus noise", `http://www.iua.upf.es/~sms/`, January 1997.
6. McAulay, M. and Quatieri. "Speech Coding and Synthesis", edited by W.B. Kleijn and K.K. Paliwal. chapter 4, Elsevier Science B.V., 1995.
7. Goodwin, M.. and Vetterli, M., "Time-frequency models for music analysis, transformation, and synthesis", Time-Frequency Time-Scale Symposium, Paris, 1996.
8. Prandoni, P., Goodwin, M., and Vetterli, M., "Optimal time segmentation for signal modeling and compression", *ICASSP-1997*.
9. Bosi, M., Brandenburg, K., *et. al*, "ISO-IEC MPEG-2 Advanced Audio Coding", 101st AES Convention, Los Angeles, 1996.
10. Anderson, D.V., "Speech analysis and coding using a multi-resolution sinusoidal transform", *ICASSP-1996*.
11. Rodriguez-Hernandez, M., and Casajus-Quiros, F., " Improving time-scale modification of audio signals using wavelets", *ICSPAT-1994*, vol. 2, pp. 1573-1577.
12. Ellis, D., and Vercoe, B., "A wavelet-based sinusoid model of sound for auditory signal separation", *International Computer Music Conference*, p.86-89, 1991.
13. Edler, B, "Aliasing reduction in sub-bands of cascaded filter banks with decimation", *Electronic Letters*, Vol. 28, No. 12, 1992, p.1104-1106.
14. Tang, B., Shen, A., Pottie, G., and Alwan, A, "Spectral analysis of subband filtered signals", *ICASSP-1995*.
15. Fliege, N. J., and Zolzer, U., "Multi-Complementary filter bank", *ICASSP-1993*.
16. Burt, P.J., and Adelson, E.A., "The laplacian pyramid as a compact image code", *IEEE Transactions on Communications*, Vol. 31, No. 4, April 1983.
17. Thomson, D., "Spectral estimation and harmonic analysis", *Proceedings of the IEEE*, Vol. 70, No. 9, Sept. 1982.
18. Verma, T., Levine, S., and Meng, T., "Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals", *International Computer Music Conference*, Greece, 1997.