SOFTWARE FOR SPECTRAL ANALYSIS, EDITING, AND SYNTHESIS

Michael Klingbeil
michael@klingbeil.com

ABSTRACT

This paper describes the design and development of new software for spectral analysis, editing and resynthesis. Analysis is accomplished using a variation of the traditional McAulay-Quatieri technique of peak interpolation and partial tracking. Linear prediction of the partial amplitudes and frequencies is used to determine the best continuations for sinusoidal tracks. A high performance user interface supports flexible selection and immediate manipulation of analysis data, cut and paste, and unlimited undo/redo. Hundreds of simultaneous partials can be synthesized in real-time and documents may contain thousands of individual partials dispersed in time without degrading performance. A variety of standard file formats are supported for the import and export of analysis data.

1. INTRODUCTION

Sinusoidal modeling has a proven track record of high quality resynthesis while offering numerous possibilities for novel sonic transformations [1, 3, 7, 8]. Software such as Lemur [4], AudioSculpt, and MetaSynth, to name but a few, have demonstrated the power and popularity of graphical user interfaces for spectral editing.

A new software application named SPEAR, Sinusoidal Partial Editing Analysis and Resynthesis, has been created to offer increased speed, flexibility, and ease of use in the domain of graphical spectral editing. The following goals were kept in mind throughout the design and development phases: editing should be as fast and as easy to understand as in a time domain waveform editor, listening to transformations should be possible with no intermediate synthesis or processing stage, and high quality analyses should require only a few parameter settings. Additionally, SPEAR interoperates well with other analysis-resynthesis software by offering both SDIF [11] and native format data exchange.

In order to offer a familiar and comfortable interface, SPEAR is written to run using the native graphics of the host operating system. Portability is made possible using wxWidgets (`http://www.wxwidgets.org`), a C++ GUI library which allows the software to be compiled for MacOS, Windows, and GTK Linux. Current builds of SPEAR run on MacOS 9, MacOS X, and Windows.

2. ANALYSIS

Audio analysis adheres to the basic peak interpolation and partial tracking methods as detailed in [7, 10]. To begin analysis the user must specify minimum frequency spacing in hertz, f_a, which for a harmonic analysis should correspond to the fundamental frequency. This is used to determine a default window length and main lobe width, as well FFT size and analysis rate (hop size). For inharmonic or polyphonic sounds, f_a will determine the minimum frequency spacing between partials that can be independently resolved. Given a window length in samples of $M = 4f_s/f_a$ (where f_s is the sampling rate), the desired FFT size is $N = 2^{\lceil \lg M \rceil + 1}$, resulting in a spectrum oversampling by a minimum factor of 2.

2.1. Peak Selection

The local maxima in the magnitude spectrum guide the search for sinusoidal peaks. Parabolic interpolation of frequency and phase [7, 10] using bins $n-1$, n, $n+1$ selected from the $N/2$ bins of an analysis frame yield frequency, phase, and amplitude estimates for the candidate peak. With a minimum mainlobe width of $4f_a$ and spectrum oversampled by a minimum factor of 2, each candidate peak will be sampled by at least 8 FFT bins. To aid the rejection of spurious peaks we may conservatively impose the restriction that the three magnitudes a_{n-1}, a_n, a_{n+1} of a parabolic peak must also all exceed the magnitude of either neighboring bin; $a_{n-1} > a_{n-2}$ or $a_{n+1} > a_{n+2}$.

2.2. Amplitude Thresholds

Two types of amplitude thresholds are used in the peak picking and partial tracking process. For a peak to be considered a candidate for tracking, it must exceed an absolute threshold T_d (default value of -96 dB). These peaks are matched with existing partial tracks as described in the next section. If no appropriate track continuation is found, the partial ends — so T_d may be considered a "death" threshold for possible continuation of a partial.

Unmatched peaks become candidates for initiating new partial tracks. A new track is started only if a peak exceeds a dynamic threshold level T_b which is computed with a frequency dependent threshold curve $A_t(f_k)$ in combination with the maximum bin amplitude of the analysis frame, $a_n^{\max}$. The curve is given by

$$A_t(f_k) = a_L - a_R + (a_R)(b^{f_k/20}) \qquad (1)$$

where f_k is the peak frequency in kHz, b is a parameter controlling the shape of the curve, a_L is the amplitude threshold at 0 kHz, and a_R is the range of the curve in positive dB from 0 to 20 kHz. Figure 1 shows $A_t(f_k)$

with default values $b = 0.0075$, $a_L = -24$, and $a_R = 32$. Given the maximum bin amplitude in dB for a frame, $a_n^{\max}$ and the frequency f_p in Hz of the peak under consideration, $T_b = a_n^{\max} + A_t(f_p/1000)$. Thus T_b may be considered a "birth" threshold for potential new partials. The birth threshold requires low-frequency peaks to exceed a greater threshold than high-frequency ones.

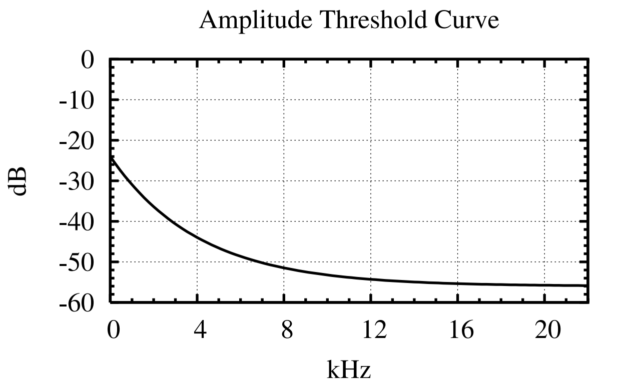


Figure 1. Default frequency dependent threshold curve

This helps to compensates for the spectral rolloff of most natural sounds and allows the analysis to find more high frequency partial tracks. This also avoids resyntheses that sound "dull" and heavily low-pass filtered. An additional benefit is improved rejection of spurious low frequency, low amplitude partial tracks.

2.3. Partial Tracking

The linear prediction method of [5] is used to match the evolving sinusoidal tracks with the detected peaks. Predicted values are computed for both the frequency and amplitude of a track. The partial tracking operates as follows: At frame i we have detected N candidate peaks and we have K active sinusoidal tracks which extend to at most frame $i - 1$. For each of K active sinusoidal tracks, the Burg method is used to compute two sets of linear prediction coefficients, one on frequency and one on amplitude. An LP model of order 6 computed from a maximum of 64 previous values has worked well in practice. It is critical that enough points are used to capture periodic features such as amplitude modulation or vibrato. The LP coefficients for track k are used to predict frequency f_k^{pr} and amplitude a_k^{pr} values for frame i. When a new sinusoidal track starts and there are not enough values to compute the LP coefficients, the mean frequency and amplitude are used as the predicted values.

The error between predicted values for track k and peak n are given by a measure of Euclidean distance,

$$E_{k,n} = \sqrt{\left[12\lg\left(\frac{f_n^{obs}}{f_k^{pr}}\right)\right]^2 + \left[\alpha\, 20\log_{10}\left(\frac{a_n^{obs}}{a_k^{pr}}\right)\right]^2} \tag{2}$$

where f_n^{obs} and a_n^{obs} are the observed frequency and amplitude values for peak n. The first term measures distance in semitones and the second in dB with a scale factor α. Informal tests have shown $\alpha = {}^1\!/_{12}$ to offer a reasonable weighting between prediction errors in frequency versus those in amplitude.

Each track k selects the best continuation peak n over all N peaks subject to constraints on the maximum allowable difference in predicted versus actual frequency. More precisely $|f_n^{obs} - f_n^{pr}| < \Delta f_{max}$ where Δf_{max} is proportional to the analysis frequency f_a. For harmonic sounds $\Delta f_{max} = \frac{3}{4} f_a$ is a reasonable default value.

Tracks which fail to find a match either become inactive or lie dormant for several successive frames. In the case of a track which has been dormant for j frames, linear prediction is used to predict the $j + 1^{\text{th}}$ values for matching to candidate peaks. This allows temporal gaps in the analysis data to be spanned.

3. DATA STORAGE

Rather than represent the analysis data as a sorted list of time frames, SPEAR uses a list of partials which are represented by breakpoint functions of time versus amplitude, frequency, and phase. With this storage model the implementation of cut, copy, and paste operations, both of entire partials or segments, is straightforward. Additionally partials can be individually shifted, stretched, or compressed in time without resampling to fixed frame time points. A further advantage is that the storage model can easily support multirate analysis data [6] or time reassigned breakpoints [3].

3.1. Time-span frames

A common operation, particularly during synthesis, is determining the list of partials (and breakpoint segments) crossing a particular time point. In the frame based storage model, a binary search of sorted frames quickly determines the partials active at any particular time.

With a list of breakpoint partials, this query requires an iteration through all the partials in the data set. For a short sound this may only require looking at several hundred elements, but for longer sounds there are likely to

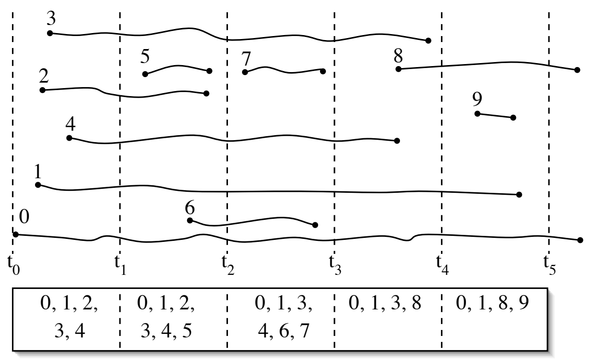


Figure 2. Data structure for division of partials into time-span frames. Comma delimited lists for each frame indicate the active partials.

be many partials of short duration. For example, the total number of partials for a one minute sound could be well over twenty-thousand. During synthesis continued iteration over thousands of partials is too inefficient. A solution is to maintain a parallel data structure of time-span frames. For each frame, a list is kept of all partials that have breakpoints within the frame's time span — typically on the order of $^1/_8$ second. Figure 2 shows an example of ten partials segmented into five frames where each frame contains a list of its active partials. The active partial lists maintain a relatively constant size as partials turn on and off. For a typical sound, this reduces active partial lookups to an iteration over only several hundred elements.

4. DATA EXCHANGE

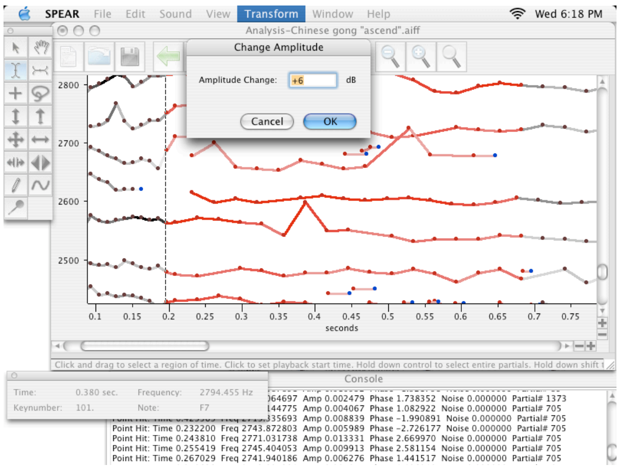
A goal of the SPEAR project is to allow data exchange with existing analysis synthesis packages. For example one might wish to use SPEAR's visualization to compare the analysis results from different software. The SDIF file format [11] is supported both for import and export. For SDIF import the supported frame types are 1TRC (sinusoidal tracks), 1HRM (harmonic sinusoidal tracks), RBEP (reassigned bandwidth enhanced partials – a matrix type used by Loris [3]), and 1STF (short-time fourier transform frames). Data can be exported either as 1TRC frames or RBEP frames. In the case of 1TRC frames, resampling of the breakpoint functions must be performed to conform to the fixed frames of the 1TRC file type. For RBEP frames, each point has a time offset value. By collating all breakpoints into frame sized chunks and properly setting the time offset, any distribution of breakpoints can be losslessly represented with an RBEP SDIF file.

Additional formats are supported for importing — the .mq and .an formats from SNDAN [1] and the .ats format from the ATS package [8]. The data exchange options are summarized in Table 1.

Two custom text file formats have been implemented for both import and export. "Text frames" represents resampled frames, where each line of the file contains the data of a single time frame. Index numbers are used to link sinusoidal tracks. "Text partials" represents exact breakpoint functions, where each line of the file represents a complete partial. The text formats ease the implementation of simple programs performing additional analysis or transformation of the data.

Format	Import	Export
SDIF 1TRC	×	×
SDIF RBEP	×	×
SDIF 1HRM	×	
SDIF 1STF	×	
SNDAN .mq	×	
SNDAN .pv	×	
ATS .ats	×	
Text frames	×	×
Text partials	×	×

Table 1. Supported file formats



```
)64697  Amp 0.002479  Phase 1.738352   Noise 0.000000  Partial# 1373
144775  Amp 0.004067  Phase 1.082922   Noise 0.000000  Partial# 705
.335693  Amp 0.008839  Phase -1.990891  Noise 0.000000  Partial# 705
Point Hit: Time 0.232200  Freq 2743.872803  Amp 0.005989  Phase -2.726177  Noise 0.000000  Partial# 705
Point Hit: Time 0.243810  Freq 2771.031738  Amp 0.013331  Phase 2.669970   Noise 0.000000  Partial# 705
Point Hit: Time 0.255419  Freq 2745.404053  Amp 0.009913  Phase 2.581154   Noise 0.000000  Partial# 705
Point Hit: Time 0.267029  Freq 2741.940186  Amp 0.006276  Phase 1.441517   Noise 0.000000  Partial# 705
```

Figure 3. Zoomed-in display showing amplitude adjustment dialog

5. USER INTERFACE

The SPEAR user interface offers speedy performance and intuitive interaction. The analysis data is displayed with time on the abscissa and frequency on the ordinate. As in a black-and-white spectrogram, varying levels of gray shading indicate the relative amplitudes of partials. Following the model of time domain waveform editors, the amount of detail shown for the breakpoint functions varies according to a user controlled zoom factor. At high zoom levels, every breakpoint is shown and clickable handles allow individual manipulation in time and frequency (Figure 3). At lower zoom levels, the decrease in detail avoids visual clutter and significantly speeds redrawing.

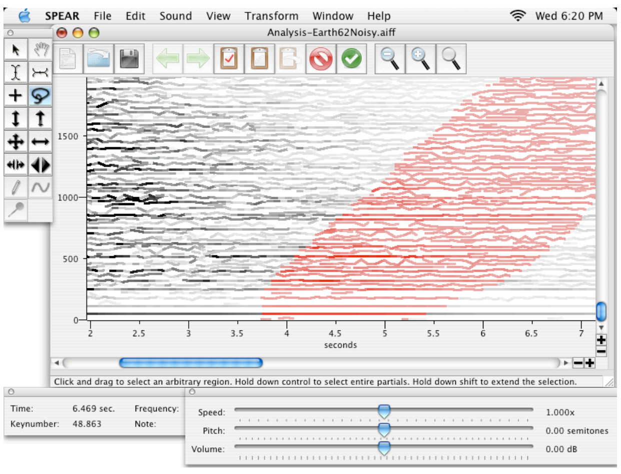
Interaction follows a selection and direct manipulation paradigm. For example, clicking on a partial selects it and shift clicking adds partials to the current selection. Choosing the transpose tool then allows the selected partials to be transposed in pitch by dragging up or down. Additional selection modes allow the user to sweep out areas in time and frequency or to draw arbitrary time-frequency regions. Following the model of graphics editors, arbitrary selections can be made up of a union or difference of individual contiguous selections.

Editing tools and commands which operate on the current selection include transposition, frequency shifting, time shifting, time stretching, and amplitude adjustment.

Unlimited undo/redo is supported for all editing operations and most editing can be performed while the sound is being synthesized. For further real-time control, sliders allow adjustment of the overall transposition level, amplitude, and playback speed.

6. SYNTHESIS

SPEAR offers several different synthesis methods. For real-time playback, the inverse FFT method [9] offers ex-


SPEAR File Edit Sound View Transform Window Help
Wed 6:20 PM
Analysis-Earth62Noisy.aiff
1500
1000
500
0
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
seconds

Click and drag to select an arbitrary region. Hold down control to select entire partials. Hold down shift to extend the selection.

Time: 6.469 sec.
Keynumber: 48.863
Frequency:
Note:

Speed: 1.000x
Pitch: 0.00 semitones
Volume: 0.00 dB

Figure 4. Zoomed-out display showing lasso selection and playback control sliders

cellent efficiency and very good quality. Although most synthesis artifacts of the standard IFFT method can be minimized with the use of appropriate overlap-add windows, extremely rapid modulations of frequency or amplitude may not be adequately reproduced. For the highest quality sound, SPEAR can perform oscillator bank synthesis with optional cubic phase interpolation as in the classical McAulay-Quatieri method. SPEAR also supports the resynthesis of Loris RBEP files which include a noise component for each breakpoint [3]. These so-called bandwidth enhanced partials can be synthesized with either the IFFT method or noise modulated oscillators.

7. CONCLUSION

The SPEAR package integrates important developments in the additive analysis-synthesis field — SDIF, IFFT synthesis, LP partial tracking — into a powerful, easy-to-use package. Although SPEAR is currently a reasonably complete tool, there are numerous capabilities that can be added.

In particular there are many additional transformation methods to implement, including cross-synthesis, retuning, and algorithmic generation of partials. Additional file formats should be supported as well, such as SDIF EASF frames, which are ideally suited for breakpoint functions [2]. To this end SPEAR needs a published C API to make plug-in modules easier to implement. A scripting layer would be a powerful addition, allowing the user to directly experiment with algorithmic transformations.

The integration of the most current developments in noise modeling, transient modeling, and multirate analysis are also important. Noise and transient modeling in particular pose particular challenges for the user interface. With some methods, noise can be associated directly with partials. In other models, separate noise bands may need to be displayed. Similarly, transients would require another display layer as well as specific constraints on the type of transformations that can be applied to them. A generalized layered display could not only display multichannel analyses, but other useful temporal data such as fundamental frequency, brightness, or spectral envelopes. It is expected that many of these ideas will be implemented in future versions. The software is available for download at `http://www.klingbeil.com/spear`.

8. REFERENCES

[1] Beauchamp, J. W. “Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds,” *Audio Engineering Society*, Preprint no. 3479, 1993.

[2] Bresson, J. and C. Agon. “SDIF Sound description data representation and manipulation in computer assisted composition,” *Proceedings of the ICMC*, Miami, USA, 2004, pp. 520-527.

[3] Fitz, K. and L. Haken. “On the Use of Time-Frequency Reassignment in Additive Sound Modeling,” *Journal of the AES*, vol. 50, no. 11, Nov. 2002, pp. 879-893.

[4] Fitz, K., L. Haken, and B. Holloway. “Lemur - A Tool for Timbre Manipulation,” ICMC, Banff Centre, Canada, 1995, pp. 158-161.

[5] Lagrange, M., S. Marchand, M. Raspaud, and J. Rault. “Enhanced Partial Tracking Using Linear Prediction,” *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK, 2003.

[6] Levine, S. *Audio Representations for Data Compression and Compressed Domain Processing*. Ph.D. Dissertation, Stanford University, Dec. 1998.

[7] McAulay, R. J. and T. F. Quatieri. “Speech Analysis/Synthesis Based on A Sinusoidal Representation,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 4, Aug. 1986, pp. 744-754.

[8] Pampin, J. “ATS: A System for Sound Analysis Transformation and Synthesis Based on a Sinusoidal plus Critical-Band Noise Model and Psychoacoustics,” *Proceedings of the ICMC*, Miami, USA, 2004, pp. 402-405.

[9] Rodet, X. and P. Depalle. “Spectral Envelopes and Inverse FFT Synthesis,” *Audio Engineering Society*, Preprint no. 3393, 1992.

[10] Smith, J. O. and X. Serra. “PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation,” *Proceedings of the ICMC*, Champaign-Urbana, USA, 1987, pp. 290-297.

[11] Wright, M., A. Chaudhary, A. Freed, S. Khoury, and D. Wessel. “Audio Applications of the Sound Description Interchange Format Standard,” *Audio Engineering Society*, Preprint no. 5032, 1999.