

Sound Morphing using Loris and the Reassigned Bandwidth-Enhanced Additive Sound Model: Practice and Applications

Kelly Fitz[†] Lippold Haken[‡] Susanne Lefvert[§] Mike O'Donnell[¶]

Abstract

The reassigned bandwidth-enhanced additive sound model is a high-fidelity representation that allows manipulations and transformations to be applied to a great variety of sounds, including noisy and non-harmonic sounds. Combining sinusoidal and noise energy in a homogeneous representation, the reassigned bandwidth-enhanced model is ideally suited to sound morphing, and is implemented in the open source software library, Loris. This paper presents methods for using Loris and the reassigned bandwidth-enhanced additive model to achieve high-fidelity sound representations and manipulations, and introduces new software tools allowing non-programmers to avail themselves of the sound modeling and manipulation capabilities of the Loris package.

1 Introduction

The Reassigned Bandwidth-Enhanced Additive Model is similar in spirit to traditional sinusoidal models (McAulay and Quatieri 1986; Serra and Smith 1990; Fitz and Haken 1996) in that a waveform is modeled as a collection of components, called *partials*, having time-varying amplitude and frequency envelopes. Our partials are not strictly sinusoidal, however. We employ a technique of *bandwidth enhancement* to combine sinusoidal energy and noise energy into a single partial having time-varying frequency, amplitude, and noisiness (or bandwidth) parameters (Fitz, Haken, and Christensen 2000a). The bandwidth envelope allows us to define a single component type that can be used to manipulate both sinusoidal and noisy parts of sound in an intuitive way. The encoding of noise associated with a bandwidth-enhanced partial is robust under time-dilation and other model-domain transformations,

and is independent of other partials in the representation.

We use the method of reassignment (Auger and Flandrin 1995) to improve the time and frequency estimates used to define our partial parameter envelopes. The breakpoints for the partial parameter envelopes are obtained by following ridges on a reassigned time-frequency surface. Our algorithm shares with traditional sinusoidal methods the notion of temporally connected partial parameter estimates, but by contrast, our estimates are non-uniformly distributed in both time and frequency. This model yields greater resolution in time and frequency than is possible using conventional additive techniques, and preserves the temporal envelope of transient signals, even in modified reconstruction (Fitz, Haken, and Christensen 2000b).

The combination of time-frequency reassignment and bandwidth enhancement yields a homogeneous model (i.e. a model having a single component type) that is capable of representing at high fidelity a wide variety of sounds, including non-harmonic, polyphonic, impulsive, robustness of the reassigned bandwidth-enhanced model make it particularly well-suited for such manipulations as cross synthesis and sound morphing.

Reassigned bandwidth-enhanced modeling and rendering and many kinds of manipulations, including sound morphing, have been implemented in an open-source software package called *Loris*. But *Loris* offers only programmatic access to this functionality, and is difficult for non-programmers to use. We begin this paper with an introduction to the selection of analysis parameters to obtain high-fidelity, manipulable representations using *Loris*, and continue with a discussion of the sound morphing algorithm used in *Loris*. Finally, we present three new software tools that allow composers, sound designers, and other non-programmers to take advantage of the sound modeling, manipulation, and morphing capabilities of *Loris*.

2 Reassigned Bandwidth-Enhanced Analysis Parameters

We have designed the reassigned bandwidth-enhanced analyzer in *Loris* to have parameters that are

[†]Department of Electrical Engineering and Computer Science, Washington State University. Email: kfitz@eecs.wsu.edu

[‡]CERL Sound Group, University of Illinois at Urbana-Champaign. Email: lippold@cerlsoundgroup.org

[§]Lulea University of Technology, and Department of Computer Science, The University of Chicago. Email: slefvert@hotmail.com

[¶]Department of Computer Science, The University of Chicago. Email: odonnell@cs.uchicago.edu

few and orthogonal. That is, we have minimized the number of parameters required, and also minimized the interaction between parameters, so that changes in one parameter would not necessitate changes in other parameters. Moreover, we have made our parameters hierarchical, so that in most cases, a good representation can be obtained by adjusting only one or two parameters, and only rarely is it necessary to adjust more than three. Consequently, and in contrast to many other additive analyzers, the parameter space of the reassigned bandwidth-enhanced analyzer in Loris is smooth and monotonic, and it is easy to converge quickly on an optimal parameter set for a given sound.

The reassigned bandwidth-enhanced analyzer can be configured according to two parameters: the instantaneous frequency resolution, or minimum instantaneous frequency separation between partials, and the shape of the short-time analysis window, specified by the symmetrical main lobe width in Hz.

The *frequency resolution* parameter controls the frequency density of partials in the model data. Two partials will, at any instant, differ in frequency by no less than the specified frequency resolution. The frequency resolution should be slightly less than the anticipated partial frequency density. For quasi-harmonic sounds, the anticipated partial frequency density is equal to the harmonic spacing, or the fundamental frequency, and the frequency resolution is typically set to 70% to 85% of the fundamental frequency. For non-harmonic sounds, some experimentation may be necessary, and intuition can often be obtained using a spectrogram tool.

The shape of the short-time analysis window governs the time-frequency resolution of the reassigned spectral surface, from which bandwidth-enhanced partials are derived. An analysis window that is short in time, and therefore wide in frequency, yields improved temporal resolution at the expense of frequency resolution. Spectral components that are near in frequency are difficult to resolve, and low-frequency components are poorly represented. A longer analysis window compromises temporal resolution, but yields greater frequency resolution. Spectral components that are near in frequency are more easily resolved, and low-frequency components are more accurately represented, but short-duration events may suffer temporal smearing, and short-duration events that are near in time may not be resolved. (See, for example, (Masri, Bateman, and Canagarajah 1997) for a discussion of issues surrounding window selection in short-time spectral analysis.)

The use of time-frequency reassignment improves the time and frequency resolution of the reassigned bandwidth-enhanced model relative to traditional short-time analysis methods (Fitz, Haken, and Christensen 2000b). Specifically, it allows us to use long (narrow in frequency) analysis windows to obtain good frequency resolution, without smearing short-

duration events. However, multiple short-duration events occurring within a single analysis window still cannot be resolved. Fortunately, the improved frequency resolution due to time-frequency reassignment also allows us to use short-duration analysis windows to analyze sounds having a high density of transient events, without greatly sacrificing frequency resolution.

The choice of *analysis window width* depends on the anticipated partial frequency density, within limits. Generally, the window width is set equal to the anticipated frequency density, or fundamental frequency, in the case of quasi-harmonic sounds. For quasi-harmonic sounds, it is rarely necessary to use windows wider than 500 Hz, although good results have been obtained using windows as wide as 800 Hz to analyze a fast bongo roll. Similarly, for very low-frequency quasi-harmonic sounds, best results are often obtained using windows as wide as 120 Hz.

All other parameters of the Loris analyzer can be configured automatically from the specification of the frequency resolution and analysis window width parameters, but they are also independently accessible and configurable.

The *frequency drift* parameter governs the amount by which the frequency of a partial can change between two consecutive data points extracted from the reassigned spectral surface. This parameter is generally set equal to the frequency resolution, but in some cases, for example, in quasi-harmonic sounds having strong noise content, the frequency of some low-energy partials may tend to occasionally “wander” away from the harmonic frequency, resulting in poor harmonic tracking. In these cases, reducing the frequency drift to, say, 0.2 times (one-fifth) the fundamental frequency may greatly improve harmonic partial tracking, which is important for manipulations such as morphing.

The *hop time* parameter specifies the time difference between successive short-time analysis window centers used to construct the reassigned spectral surface. Data is generally obtained from each analysis window for all partials active at the time corresponding to the center of that window, so the hop time controls, to some degree, the temporal density of the analysis data (though, thanks to the use of time-frequency reassignment, it controls the temporal resolution of the data to a much lesser degree). The hop time used by the reassigned bandwidth-enhanced analyzer in Loris is derived from the analysis window width according to a heuristic for short-time Fourier analysis described by Allen (Allen and Rabiner 1977). In many cases, it is possible to increase the hop time, thereby reducing the volume of data, by a factor of two without compromising the quality of the representation. In other cases, it may be desirable to decrease the hop size, though the authors have never encountered such a situation.

3 Sound Morphing in Loris

Sound morphing, or timbral interpolation using traditional additive sound models is conceptually straightforward. For quasi-harmonic sounds, in which each harmonic is represented by a single sinusoidal partial, the time-varying frequencies and amplitudes of the quasi-harmonic partials in the morphed sound can be obtained by a weighted interpolation of the time-varying frequencies and amplitudes of corresponding partials in the source sounds (Dodge and Jerse 1997). This process is illustrated in Figure 1.

In Loris, though the process of partial construction is different, the morphing process is fundamentally similar. Sound morphing is achieved by interpolating the time-varying frequencies, amplitudes, and bandwidths of corresponding partials obtained from re-assigned bandwidth-enhanced analysis of the source sounds. Three independent morphing envelopes control the evolution of the frequency, amplitude, and bandwidth, or noisiness of the morph.

The description of sounds as “quasi-harmonic” implies a natural correspondence between partials having the same harmonic number. Even very noisy quasi-harmonic sounds, which, in traditional sinusoidal models, are represented by many short, jittery partials in noisy spectral regions, can be represented by a single partial for each harmonic using the reassigned bandwidth-enhanced additive model (Haken, Fitz, and Christensen 2002). This property of the model greatly simplifies the morphing process in Loris and improves the fidelity of morphs for such sounds.

For non-harmonic or polyphonic sounds, however, there may be no obvious correspondence between partials in the source sounds, or there may be many possible correspondences. Loris provides mechanisms for explicitly establishing correspondences between source partials.

3.1 Establishing Partial Correspondences

Correspondences between partials in the source sounds are established by *channelizing* and *distilling* partial data for the individual source sounds. Partial in each source sound are assigned unique identifiers, or labels, and partials having the same label are morphed by interpolating their frequency, amplitude, and bandwidth envelopes according to the corresponding morphing function. The product of a morph is a new set of partials, consisting of a single partial for each label represented in any of the source sounds.

In Loris, *channelization* is an automated process of labeling the partials in an analyzed sound. Partial can be labeled one by one, but analysis data for a single sound may consist of hundreds or thousands of partials. If the sound has a known, simple frequency structure, an automated process is much more efficient.

Channelized partials are labeled according to their

adherence to a harmonic frequency structure with a time-varying fundamental frequency. The frequency spectrum is partitioned into non-overlapping channels having time-varying center frequencies that are harmonic (integer) multiples of a specified reference frequency envelope, and each channel is identified by a unique label equal to its harmonic number. The reference (fundamental) frequency envelope for channelization can be constructed explicitly, point by point, or constructed automatically by tracking a long, high-energy partial in the analysis data. Each partial is assigned the label corresponding to the channel containing the greatest portion of its (the partial’s) energy.

The sound morphing algorithm described above requires that partials in a given source be labeled *uniquely*, that is, no two partials can have the same label. In Loris, *distillation* is the process for enforcing this condition. All partials identified with a particular channel, and therefore having a common label, are distilled into a single partial, leaving at most a single partial per frequency channel and label. Channels that contain no partials are not represented in the distilled partial data. Partial that are not labeled, that is, partials having label 0, are unaffected by the distillation process. All unlabeled partials remain unlabeled and unmodified in the distilled partial set.

Note that, due to the symmetry of the frequency channels used employed by the Loris channelizer, a frequency region below half the reference (fundamental) channel frequency is not covered by any channel, and therefore partials concentrated at frequencies far below the reference frequency envelope will remain unlabeled after channelization. In practice, few partials, if any, are found in this region.

Labeled and distilled sets of partials are morphed by interpolating the envelopes of corresponding partials according to specified morphing functions. Partial in one distilled source that have no corresponding partial in the other source(s) are crossfaded according to the morphing function (Tellman, Haken, and Holloway 1995). Source partials may also be unlabeled, or assigned the label 0, to indicate that they have no correspondence with other sources in the morph. All unlabeled partials in a morph are crossfaded according to the morphing function.

When there is no temporal overlap of partials in a frequency channel, then distillation is simply a process of linking partials end to end, and inserting silence between the endpoints. When partials in a frequency channel overlap temporally, then an algorithm is needed to determine a single frequency, amplitude, and noisiness value for the distilled partial at times in the overlap region. In Loris, the distiller resolves overlap issues by choosing the strongest (i.e. having the most energy) of the overlapping partials to construct the distilled partial. The energy in the rejected partials is not lost, rather, it is added to the distilled partial as noise energy in a process called *energy redistribution*.

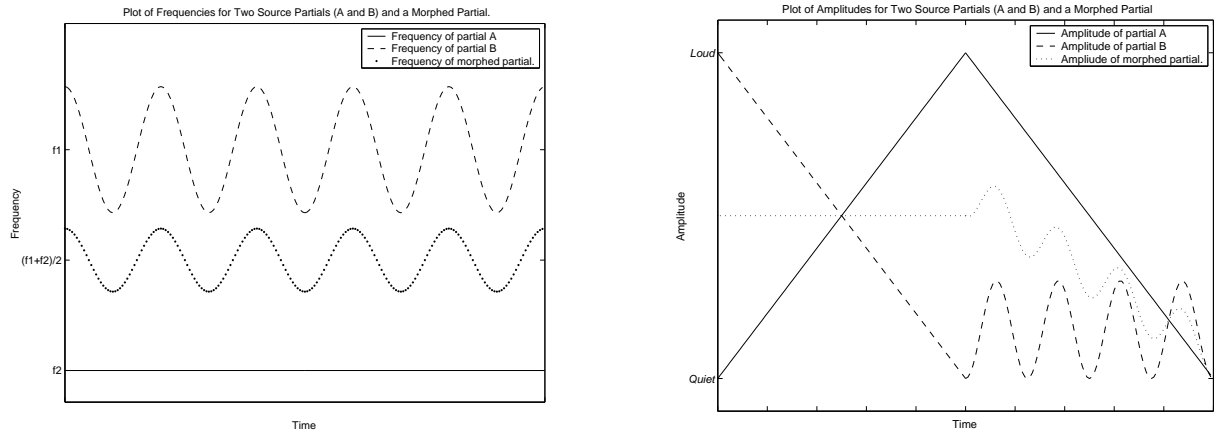


Figure 1: Equal-weight morphing of a hypothetical pair of partials by interpolation of their frequency (left plot) and amplitude (right plot) envelopes. The source partial envelopes are plotted with solid and dashed lines, and the envelopes corresponding to a 50% (equal-weight) morph are plotted with dotted lines. Note that these envelopes are artificially generated to illustrate the morphing operation, and do not correspond to any real sounds.

(Fitz, Haken, and Christensen 2000a).

In some cases, the energy redistribution effected by the distiller is undesirable. In such cases, the partials can be *sifted* before distillation. The *sifting* process in Loris identifies all the partials that would be rejected (and converted to noise energy) by the distiller and assigns them a label of 0. These sifted partials can then be identified and treated separately or removed altogether, or they can be passed through the distiller unlabeled, and crossfaded in the morphing process.

The various morph sources need not be distilled using identical sets of frequency channels. However, dramatic partial frequency sweeps will dominate other audible effects of the morph, so care must be taken to coordinate the frequency channels used in the distillation process. Though the harmonic frequency structure described by the channelization process may not be a good representation of the frequency structure of a particular sound (as in the case of a non-harmonic bell sound for example), it may still yield good morphing results by labeling partials in such a way as to prevent dramatic frequency sweeps.

3.2 Temporal Feature Alignment

Significant temporal features of the source sounds must be synchronized in order to achieve good morphing results. Many sounds, particularly, many familiar monophonic, quasi-harmonic sounds share a temporal structure that includes such features as attack, sustain, and release. Additionally, many such sounds have recurring temporal features such as vibrato and tremolo cycles. A morph of such sounds may be unsatisfying if the sources have very different temporal feature sets, or different numbers of temporal features, or if related temporal features (such as the end of the attack, or the beginning of the release) occur at different times.

Moreover, when synchronizing a sound morph with

a visual sequence, such as a computer animation, temporal features of the morphed sound must be aligned with visual events in the animation in order to make the relationship between sound and visuals believable or seemingly “natural” (Bargar et al. 2000).

Loris provides a *dilation* mechanism for non-uniformly expanding and contracting the partial parameter envelopes to redistribute temporal events. For example, when morphing instrument tones, it is common to align the attack, sustain, and release portions of the source sounds by dilating or contracting those temporal regions.

The process of resolving conflicts between different numbers of temporal features, such as different numbers of vibrato cycles, is beyond the scope of this paper, but has been addressed in (Tellman, Haken, and Holloway 1995).

This process of time dilation can occur before or after distillation, but is an essential component in controlling the evolution of the morph.

3.3 Other Deformations

Deformation and temporal dilation of the partial parameter envelopes can be applied as needed at any point in the morphing process. The reassigned bandwidth-enhanced additive model is highly robust under such transformations (Fitz, Haken, and Christensen 2000a). Since the product of the morphing process in Loris is a set of partials like any other, it can be further deformed or manipulated in any of its parameters (time, frequency, amplitude, and noisiness), or morphed with yet another source sound, to achieve *N*-way morphs.

For example, quasi-harmonic sounds of different pitches may be pitch-aligned (shifted to a common pitch) before morphing, and then the morphed partials may be shifted again to a desired pitch. (Of course, in this example, the desired pitch could also have been

chosen as the common pitch before the morph.)

In some cases, the dramatic effect of a morph, and its apparent “realism” are enhanced by applying frequency or amplitude deformations that are synchronous with the evolution of the morph. This enhanced realism is particularly important when the sound morph is to be coupled with a visual presentation. In a computer animation, *Toy Angst*, slight pitch and amplitude deformations were applied to morphed sounds to accentuate the spasms of a child’s squeaky ball as it was deformed into toys of other shapes. These were found to greatly increase the realism of the presentation, and the fusion of the audio and visual morphs into a single percept (Bargar et al. 2000).

4 Applications

The Loris software consists of a C++ class library, a C-linkable procedural interface, interface files that allow Loris extension modules to be built for a variety of scripting languages using David Beazley’s Simplified Wrapper Interface Generator (SWIG) (available at <http://www.swig.org/>) (Beazley 1998), and standard UNIX/Linux tools that build and install the Loris library, headers, and extension modules for Python and Tcl. Loris is distributed as free software under the GNU General Public License (GPL), and is available at the Loris web site, <http://www.cerlsoundgroup.org/Loris/>.

The Loris software provides only programmatic access to the reassigned bandwidth-enhanced sound model and to the morphing and manipulation functionality described in Section 3, and is therefore usable only with difficulty by non-programmers. Recently, however, several software tools have been developed that allow non-programmers to access the powerful sound modeling, morphing, and manipulation features in Loris.

4.1 Fossa

To bridge the gap between sound designers and computer programmers, a graphical control application, called *Fossa*, is under development and distributed as part of the Loris project.

Fossa includes both a graphical representation of reassigned bandwidth-enhanced analysis data and the ability to audition sounds rendered from such data, allowing a user to see and hear the results of different manipulations. Reassigned bandwidth-enhanced partials can be imported from Spectral Description Interchange Format (SDIF) files (Wright et al. 2000), which are supported by Loris. Alternatively, Fossa can perform reassigned bandwidth-enhanced analysis of AIFF-format samples files, according to user-specified analysis parameters.

Imported partials are displayed in amplitude, frequency, and noise plots. The parameter envelopes for a

collection of imported partials are plotted against time in distinct amplitude, frequency, and noisiness graphs. Several sounds can be imported into Fossa at once, and manipulations applied to the displayed partial collection (selected from a pop-up menu) are immediately reflected in the parameter envelope plots. Manipulations available in Fossa include parameter scaling operations (such as pitch shifting) as well as channelization and distillation in preparation for morphing. This interactive graphical representation makes it possible to visualize the effects of complex operations like distillation, and to evaluate the suitability of various channelization strategies. Fossa’s parameter envelope display is shown in Figure 2.

Fossa provides a graphical interface for interactive construction and application of morphing control functions. Independent breakpoint envelopes for morphing frequency, amplitude, and noisiness can be assembled and manipulated in the click-and-drag editor shown in Figure 3, and applied using the Loris morphing algorithms. Finally, a morphed or otherwise manipulated sound can be rendered from reassigned bandwidth-enhanced partial data for audition or export to a samples file.

Fossa is developed by Susanne Lefvert, as a Master’s Thesis for Lulea University of Technology in collaboration with University of Chicago. Fossa will be released as free software under the GNU General Public License (GPL), and distributions of the beta release for UNIX and Linux operating systems will be available at the Loris web site.

4.2 Real-Time Synthesis in Kyma

Together with Kurt Hebel of Symbolic Sound Corporation, we have implemented a stream-based real-time bandwidth-enhanced synthesizer using the Kyma Sound Design Workstation (Hebel and Scaletti 1994).

Many real-time synthesis systems allow the sound designer to manipulate streams of samples. In our real-time bandwidth-enhanced implementation, we work with streams of data that are not time-domain samples. Rather, our *Envelope Parameter Streams* encode frequency, amplitude, and bandwidth envelope parameters for each bandwidth-enhanced partial (Haken, Fitz, and Christensen 2002; Haken, Tellman, and Wolfe 1998).

Much of the strength of systems that operate on sample streams is derived from the uniformity of the data. This homogeneity gives the sound designer great flexibility with a few general-purpose processing elements. A wide variety of real-time manipulations on envelope parameter streams, including frequency shifting, formant shifting, time dilation, cross synthesis, and sound morphing, have been implemented in the Kyma environment.

The Kyma data flow graph shown in Figure 4 produces a real-time timbre morph between two sounds

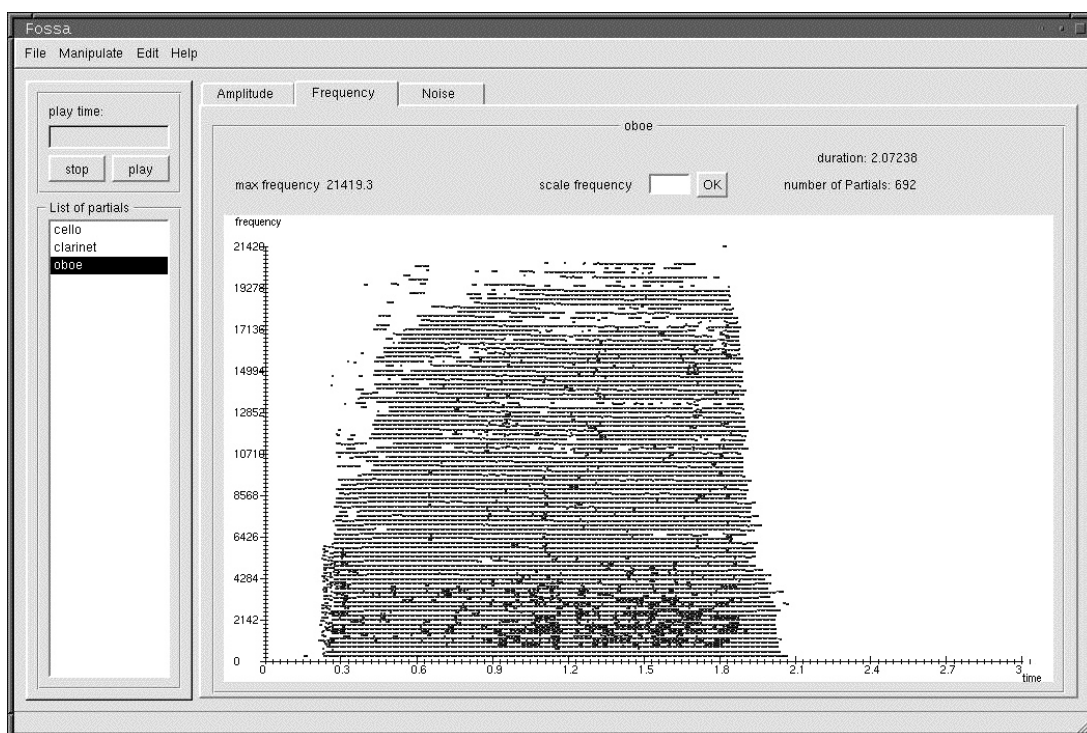


Figure 2: Parameter envelope display in Fossa showing frequency envelopes for partials in a reassigned bandwidth-enhanced analysis of an oboe tone.

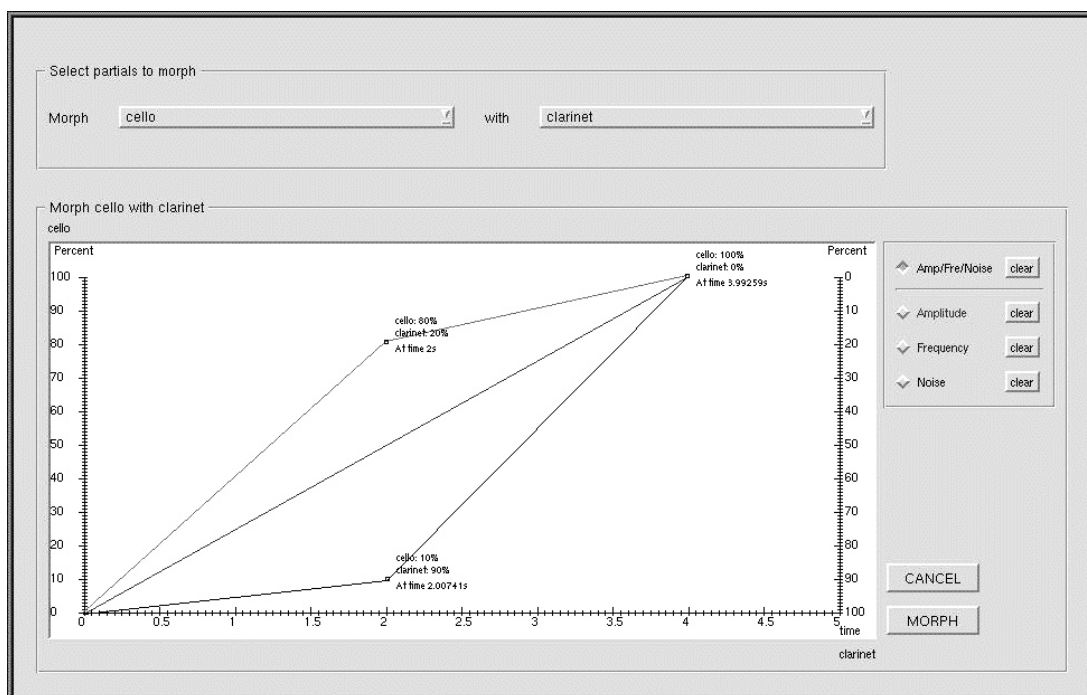


Figure 3: Morphing function editor in Fossa, showing individual frequency, amplitude, and noisiness morphing functions for a morph between a cello and a clarinet.

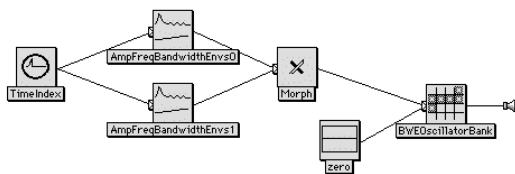


Figure 4: Data flow graph producing a real-time timbre morph between two sounds in the Symbolic Sound Kyma environment from reassigned bandwidth-enhanced analysis data prepared in Loris.

analyzed in Loris. Figure 4 depicts six interconnected Kyma Sound Objects and a speaker output. The top-level Sound Objects are shown, but not the parameters or internal structure of each Sound Object. Real-time data flow is from the left to right.

The leftmost Sound Object, called “TimeIndex,” is an oscillator. It is configured to generate a 0.5 Hz ramp waveform. The next two Sound Objects, called “AmpFreqBandwidthEnvs,” each read amplitude, frequency, and bandwidth envelopes from reassigned bandwidth-enhanced analysis data files prepared by Loris. Their input (the ramp wave) is a time index into the analysis data, and controls the progression of the morph through the analysis data sets. In this example, the morph progresses linearly through both analysis data sets, from beginning to end, over a duration of two seconds.

Each of the “AmpFreqBandwidthEnvs” generates an envelope parameter stream. The two envelope parameter streams feed into the “Morph” Sound Object, which computes a weighted average of the two streams to produce a single morphed envelope parameter stream. The rightmost Sound Object, called “BWEOscillatorBank,” is a bank of bandwidth-enhanced sinusoidal oscillators. Each oscillator synthesizes a single bandwidth-enhanced partial in the morphed sound, and all oscillators are evaluated each sample time. The output of the oscillator bank, the time-domain sum of its oscillators, is sent to the speaker.

The Kyma data flow graph in Figure 5 produces ten simultaneous morphs between eight source timbres, with each morph controlled by one of the fingers pressing on a *Continuum Fingerboard*. The *Continuum Fingerboard* is our new MIDI controller that allows continuous control over each note in a performance. It resembles a traditional keyboard in that it is approximately the same size and is played with ten fingers (Haken, Tellman, and Wolfe 1998). Like keyboards supporting MIDI’s polyphonic aftertouch, it continually measures each finger’s pressure. The *Continuum Fingerboard* also resembles a fretless string instrument in that it has no discrete pitches; any pitch may be played, and smooth glissandi are possible. It tracks in three

dimensions the position for each finger pressing on the playing surface. These continuous three-dimensional outputs are a convenient source of control parameters for real-time manipulations on envelope parameter streams.

The eight “AmpFreqBandwidthEnvs” in Figure 5 read amplitude, frequency, and bandwidth envelopes from eight reassigned bandwidth-enhanced analysis data files prepared by Loris. The eight envelope parameter streams they generate are scaled according to the three-dimensional position of the associated finger pressing on the *Continuum Fingerboard*. Thus, the finger position interactively controls the timbral evolution of the sound. Scaled envelope parameter streams are summed to produce a single stream for one finger. The streams for all ten fingers are combined by “Concentrator” and then synthesized by the “BWEOscillatorBank” Sound Objects. The polyphonic result is sent to the speaker.

4.3 Using Loris with Csound

A set of unit Csound generators supporting modified synthesis and morphing of reassigned bandwidth-enhanced model data is under development.

Csound is a flexible and extensible orchestra/score system in the style of “Music N” languages, and is one of the most popular and widely-distributed software synthesis applications (Boulanger 2000). Csound supports a wide variety of synthesis techniques, including analysis-based techniques, such as the phase vocoder (Dolson 1986) and linear predictive coding.

Csound unit generators for importing and manipulating reassigned bandwidth-enhanced analysis data will provide Csound’s huge user community with the sound morphing and manipulation capabilities of Loris. These tools will enable Csound users to integrate high-fidelity sound morphing and transformation into their own compositions and sound designs, and will further allow Loris users to avail themselves of the rich set of control structures and sound design tools available in Csound.

Early versions of the Loris unit generators for Csound will be available in May 2002.

5 Conclusion

The reassigned bandwidth-enhanced analyzer implemented in the Loris software library supports high-fidelity, robust modeling of a wide variety of sounds. The analyzer has a small set of orthogonal parameters that are easily tuned to arrive at an optimal configuration for a particular sound. Loris also provides many sound manipulations and transformations in the domain of the reassigned bandwidth-enhanced model, including sound morphing. Previously accessible only through programmatic interfaces (C/C++ and various scripting languages), a variety of new software tools

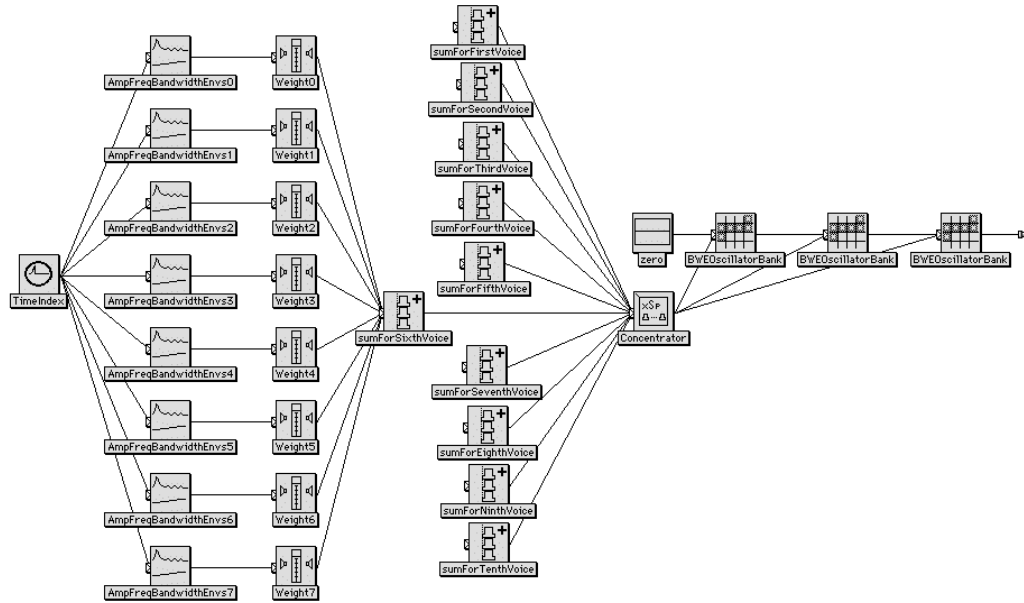


Figure 5: Data flow graph producing ten (10) independent 8-way morphs (morphs between eight source timbres) in real time in the Symbolic Sound Kyma environment controlled from a Continuum Fingerboard.

have been presented that make the sound modeling and morphing capabilities of Loris available to composers, sound designers, and other non-programmers.

References

- Allen, J. B. and L. R. Rabiner (1977, November). A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE* 65(11), 1558 – 1564.
- Auger, F. and P. Flandrin (1995, May). Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Transactions on Signal Processing* 43(5), 1068 – 1089.
- Bargar, R., A. Betts, I. Choi, and K. Fitz (2000). Models and deformations in procedural synchronous sound for animation. In *Proc. ICMC*, Berlin, Germany, pp. 205–208.
- Beazley, D. M. (1998, February). SWIG and automated C/C++ scripting extensions. *Dr. Dobbs Journal* 282, 30–36.
- Boulanger, R. (Ed.) (2000). *The Csound Book*. Cambridge, MA: MIT press.
- Dodge, C. and T. A. Jerse (1997). *Computer Music: Synthesis, Composition, and Performance* (2nd ed.). New York: Shirmer Books.
- Dolson, M. (1986). The phase vocoder: A tutorial. *Computer Music Journal* 10(4), 14 – 27.
- Fitz, K. and L. Haken (1996). Sinusoidal modeling and manipulation using Lemur. *Computer Music Journal* 20(4), 44 – 59.
- Fitz, K., L. Haken, and P. Christensen (2000a). A new algorithm for bandwidth association in bandwidth-enhanced additive sound modeling. In *Proc. ICMC*, Berlin, Germany, pp. 384–387.
- Fitz, K., L. Haken, and P. Christensen (2000b). Transient preservation under transformation in an additive sound model. In *Proc. ICMC*, Berlin, Germany, pp. 392–395.
- Haken, L., K. Fitz, and P. Christensen (to appear, 2002). Beyond traditional sampling synthesis: Real-time timbre morphing using additive synthesis. In J. W. Beauchamp (Ed.), *Sound Of Music: Analysis, Synthesis, And Perception*. Springer-Verlag.
- Haken, L., E. Tellman, and P. Wolfe (1998). An indiscrete music keyboard. *Computer Music Journal* 22(1), 30 – 48.
- Hebel, K. and C. Scaletti (1994). A framework for the design, development, and delivery of real-time software-based sound synthesis and processing algorithms. *Audio Engineering Society Preprint A-3*(3874).
- Masri, P., A. Bateman, and N. Canagarajah (1997). A review of time-frequency representations with applications to sound/music analysis-resynthesis. *Organised Sound* 2(3), 193 – 205.
- McAulay, R. J. and T. F. Quatieri (1986, August). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-34*(4), 744 – 754.
- Serra, X. and J. O. Smith (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal* 14(4), 12 – 24.
- Tellman, E., L. Haken, and B. Holloway (1995, September). Timbre morphing of sounds with unequal numbers of features. *Journal of the Audio Engineering Society* 43(9), 678 – 689.
- Wright, M., J. Beauchamp, K. Fitz, X. Rodet, A. Robel, X. Serra, and G. Wakefield (2000). Analysis/synthesis comparison. *Organised Sound* 5(3), 173 – 189.